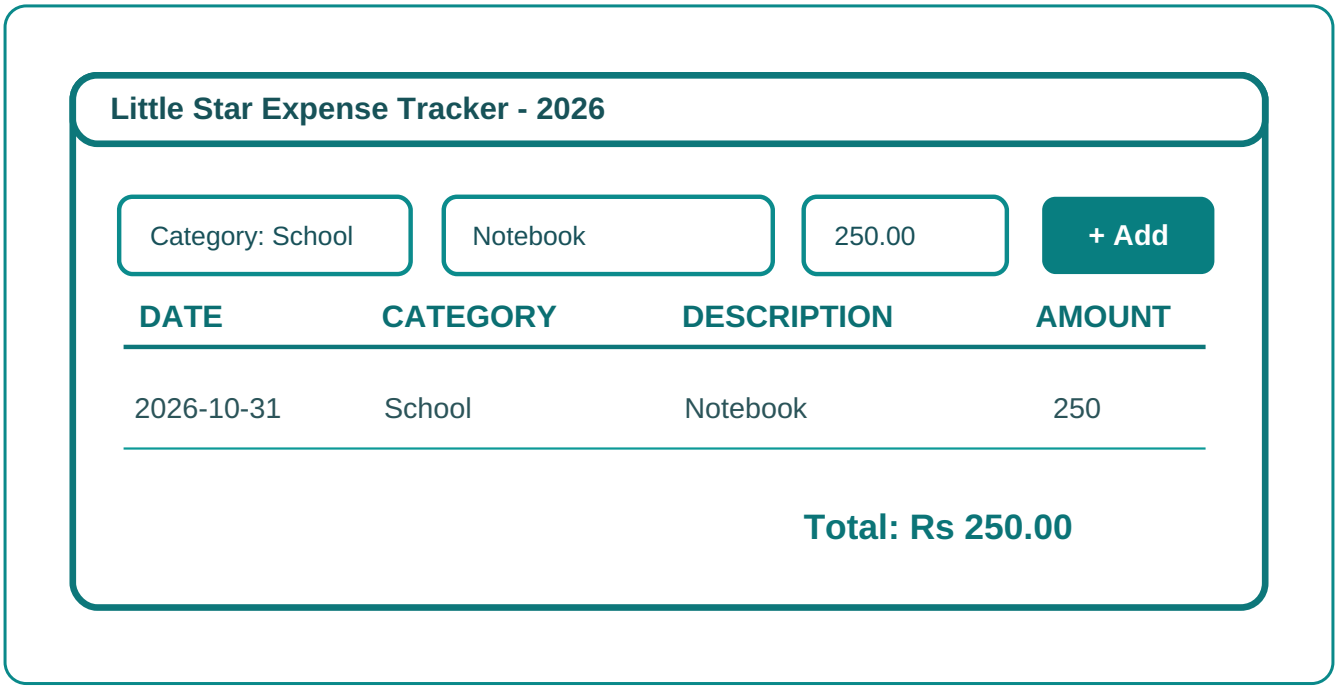
.NET සහ C#
මුල සිට ඉගෙනගනිමු

Software Development Basics • සිංහල

Windows Desktop Expense Tracker එකක් පියවරෙන් පියවර නිර්මාණය කරමු.



White Pages | C# | .NET 10 LTS | WinForms | Testing & QA

අවුරුදු 10+ ඉගෙනුම්කරුවන් සහ beginners සඳහා • Code comments English • සරල සිංහල පැහැදිලි කිරීම්

START HERE

මේ පොත භාවිතා කරන්නේ කොහොමද?

පළමුව සිංහල පැහැදිලි කිරීම කියවන්න. ඊළඟට black code panel එකේ code එක හෙමින් type කරන්න. CMD output එක බලලා පිටුවේ අභ්‍යාසය උත්සාහ කරන්න.

අපේ පොරොන්දුව

- ✓ Code copy කරනවාට වඩා, code එකේ අර්ථය පැහැදිලි කරගන්න.
- ✓ ගැටලුවක් ආවොත් error message එක කියවා bug එක හොයන්න.
- ✓ Fake/example values පමණක් භාවිතා කරන්න.
- ✓ PDF එකේ window previews ඉගෙනීම සඳහා නිර්මාණය කළ sketches බව මතක තබාගන්න.

Windows Command Prompt / CMD

```
C:\> dotnet --version
10.0.xxx
```

මුල්ම ජයග්‍රහණය

“Hello, world!” run වූ පසු ඔබේ පළමු C# project එක ඔබ විසින්ම වෙනස් කළ හැකි බව තේරෙයි.

BEFORE YOU START

අවශ්‍ය software සහ උපකරණ

Windows පරිගණකයක්, .NET 10 SDK සහ Visual Studio 2026 Community (හෝ VS Code + C# extension) භාවිතා කරනවා. Windows desktop project එකට Windows OS අවශ්‍යයි.

- ✓ Windows 11 සහ නිල .NET SDK 10
- ✓ Visual Studio 2026 හි .NET desktop development workload
- ✓ අලුත් folder එකක් සහ ප්‍රායෝගික කල්පිත දත්ත
- ✓ README සහ source ZIP එක පරිගණකයේ unzip කරන්න

Version note

2026 ඔක්තෝබර් 31 දින වන විට .NET 10 LTS stable release එකයි. .NET 11 තවම RC අවධියකයි. නිල Microsoft support policy/reference links අවසාන පිටුවල ඇත.

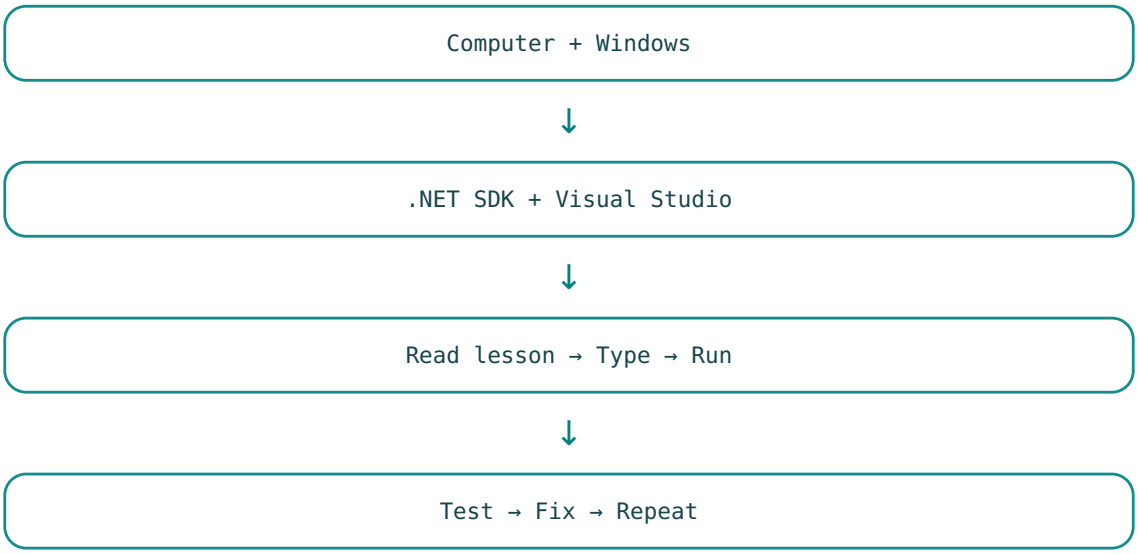


TABLE OF CONTENTS

පාඩම් අන්තර්ගතය

01 / .NET කියන්නේ මොකක්ද? 6 - 15

.NET ලෝකයට පළමු පියවර • C# සහ .NET දෙකේ වෙනස • 2026 සඳහා .NET 10 LTS • Code → Program ගමන …

02 / Setup හා Hello World 16 - 25

Windows සහ Install සැලැස්ම • Visual Studio 2026 ආරම්භය • CMD එක විවෘත කරමු • Folder එකක් සකස් කරමු …

03 / C# Programming මුල සිට 26 - 57

Statement එකක් • Variable කියන පෙට්ටිය • String සමඟ වචන • int සහ සම්පූර්ණ ගණන් …

04 / Data සහ Files 58 - 67

Data model සැලසුම • Guid යනු unique ID • JSON ආකෘතිය • Serialization …

05 / Software Engineering සහ Documentation 68 - 79

වැඩ පටන්ගන්න කලින් ප්‍රශ්නය • SDLC අදියර • SRS කියන්නේ මොකක්ද? • Functional vs Nonfunctional …

Practice project

ZIP source එකේ `START_HERE_SINHALA.md` file එකෙන් actual Windows setup instructions සොයාගන්න.

TABLE OF CONTENTS

පාඩම් අන්තර්ගතය

06 / Little Star Core Project 80 - 93

Expense record නිර්මාණය • Category තෝරන්න • Manager කියන මැදිහත්කරු • Add method flow …

07 / Windows CMD Expense App 94 - 101

Console project එක Core වෙත සම්බන්ධ කිරීම • Console menu එක • User date input • Console Add flow …

08 / Windows Forms Desktop App 102 - 118

WinForms project එක • Program.cs startup • Form එක යනු window එක • Label සහ TextBox …

09 / Testing, QA, Release 119 - 134

Testing සහ QA අතර වෙනස • Test case ලියමු • Unit test • AAA pattern …

10 / පුංචි අභ්‍යාස සහ Mini Projects 135 - 148

Lab 01 - Hello Name • Lab 02 - Pocket money • Lab 03 - Category counter • Lab 04 - Month report …

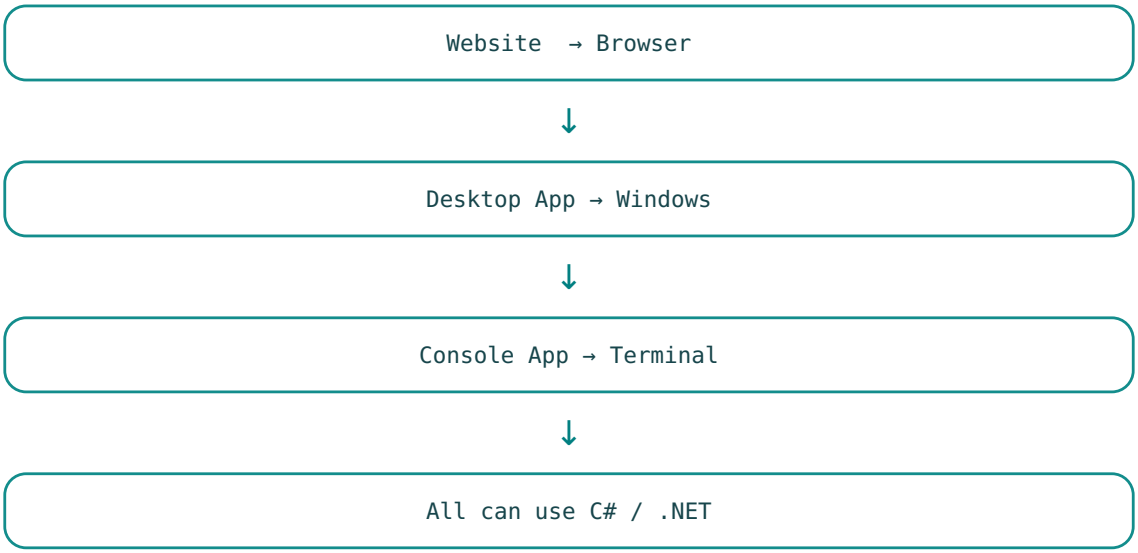
Practice project

ZIP source එකේ `START_HERE_SINHALA.md` file එකෙන් actual Windows setup instructions සොයාගන්න.

LESSON 001 / 143

.NET ලෝකයට පළමු පියවර

අපි දවසකට computer එකෙන් බොහෝ වැඩ කරනවා. ඒ වැඩවලට අවශ්‍ය software වෙනත් අය විසින් code ලියලා හදනවා. .NET කියන්නේ එවැනි apps නිර්මාණය කිරීමට Microsoft සහ open-source community එක ලබා දෙන platform එකක්.



ඇයි මෙය වැදගත්?

JavaScript browser එකකට සීමා නොවෙන විදිහට, C# භාවිතයෙන් console, desktop සහ server applications සාදන්න පුළුවන්.

පුංචි අභ්‍යාසය

ඔබ භාවිතා කරන apps තුනක් ලියා, ඒවා website, desktop හෝ mobile යන කාණ්ඩවලට බෙදන්න.

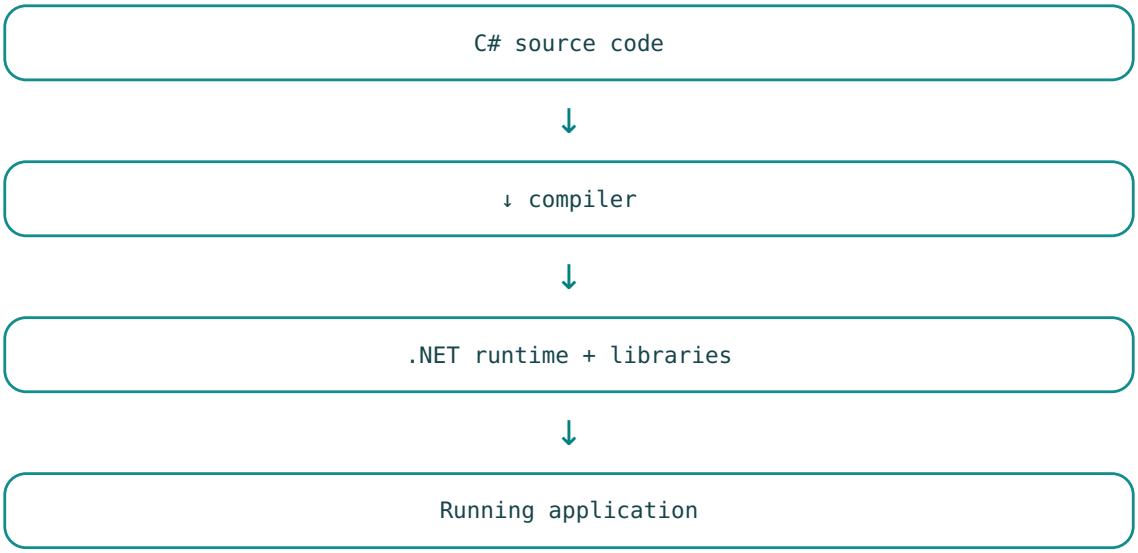
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

හොඳ පිළිතුරක්: Browser - search engine; Desktop - calculator; Mobile - alarm.

LESSON 002 / 143

C# සහ .NET දෙකේ වෙනස

C# කියන්නේ අපි instructions ලියන programming language එක. .NET කියන්නේ ඒ code compile කිරීමට, run කිරීමට සහ libraries භාවිතා කිරීමට උපකාර කරන platform එක.



ඇයි මෙය වැදගත්?

වචන ලියන භාෂාව සහ ඒ වචන ක්‍රියාත්මක කරන උපකරණ දෙක වෙනස් බව මතක තබාගන්න.

පුංචි අභ්‍යාසය

පහත වාක්‍යය සම්පූර්ණ කරන්න: C# = _____; .NET = _____.

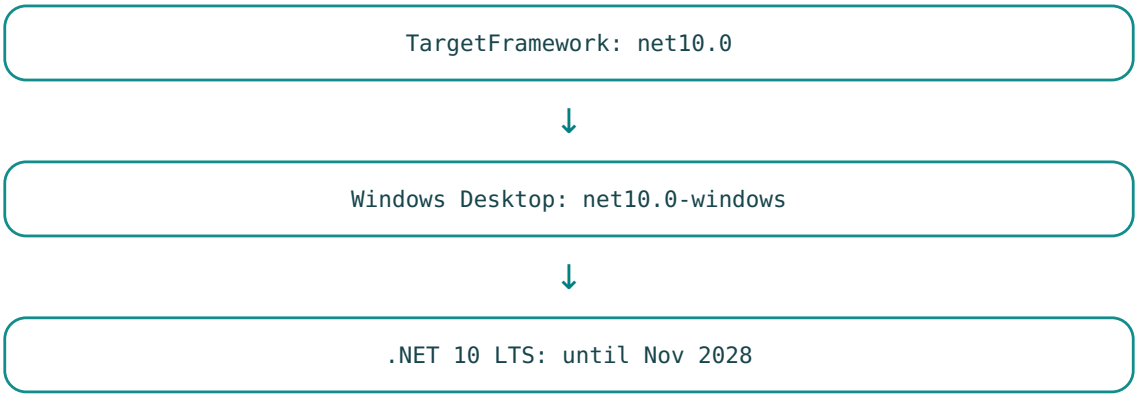
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

C# = programming language; .NET = platform.

LESSON 003 / 143

2026 සඳහා .NET 10 LTS

2026 ඔක්තෝබර් වනවිට .NET 10 යනු නිල Long Term Support (LTS) release එකකි. පාඩමට එය භාවිතා කරනවා. .NET 11 තවම release-candidate තත්ත්වයේ තිබෙන නිසා stable beginners project එකට .NET 10 තෝරනවා.



ඇයි මෙය වැදගත්?

නිල support සහ security updates ලැබෙන framework version එකක් තෝරාගැනීම developer කෙනෙකුගේ වගකීමක්.

පුංචි අභ්‍යාසය

LTS කියන අකුරු තුනේ පූර්ණ අර්ථය ලියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Long Term Support.

LESSON 004 / 143

Code → Program ගමන

ඔබ code ලියන ගොනුව `.cs` නමින් සුරකිනවා. Build command එක code compile කර app එකට අවශ්‍ය දේ සකස් කරනවා. Run කළ විට .NET runtime එක instructions ක්‍රියාත්මක කරනවා.

```
Program.cs → Build → .NET runtime → Screen
```

ඇයි මෙය වැදගත්?

Error එක compile time එකේද run time එකේද කියා වෙන් කිරීම debugging සඳහා උපකාරී.

පුංචි අභ්‍යාසය

Compile error එකක් සහ runtime error එකක් වෙන වෙනම උදාහරණ දෙකක් සිතන්න.

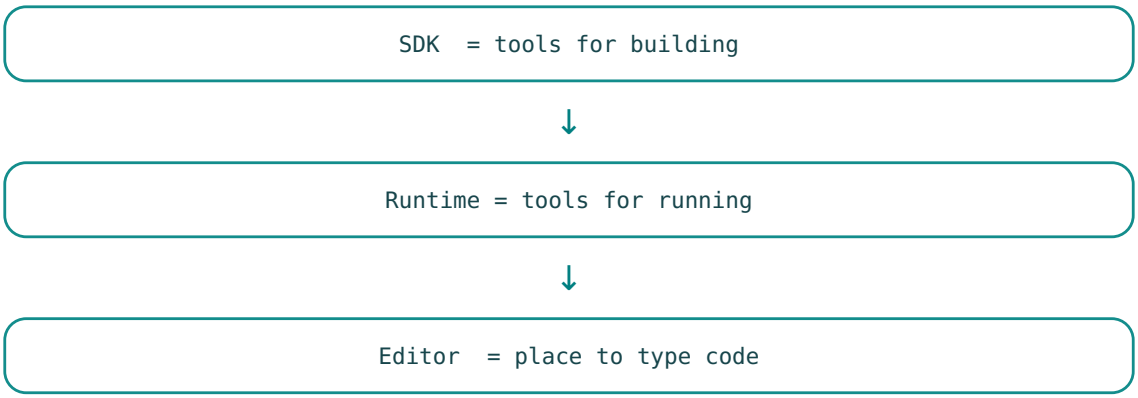
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Missing semicolon = compile-time; bad input = runtime.

LESSON 005 / 143

SDK සහ Runtime

SDK කියන්නේ software හදන කෙනාගේ tools පෙට්ටිය. Runtime කියන්නේ සකස් කළ app එක ධාවනය කරන උපකරණ කොටස. SDK install කළාම build සහ run දෙකම කළ හැකියි.



ඇයි මෙය වැදගත්?

Developer පරිගණකයක SDK අවශ්‍යයි; අවසාන පරිශීලකයාට deployment එක අනුව runtime අවශ්‍ය විය හැකියි.

පුංචි අභ්‍යාසය

SDK එක සහ editor එක අතර වෙනස පැහැදිලි කරන්න.

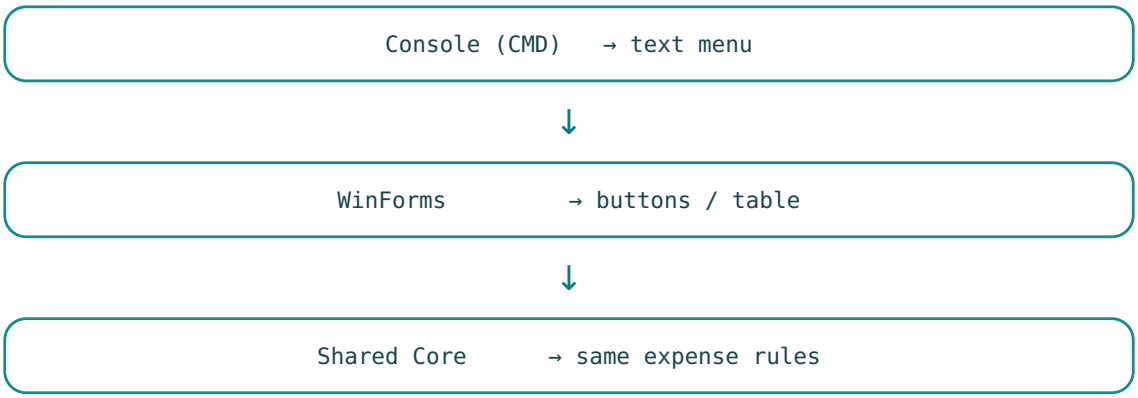
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

SDK builds code; editor helps write code.

LESSON 006 / 143

CLI, GUI සහ Web App

CLI app එකට text commands දෙනවා. GUI app එකේ buttons හා input boxes තියෙනවා. Web app එක browser එකෙන් බලනවා. අපේ project එකට CLI සහ GUI දෙකම නිර්මාණය කරනවා.



ඇයි මෙය වැදගත්?

එකම business rules දෙකටම බෙදා දීමෙන් duplicate code අඩු වෙනවා.

පුංචි අභ්‍යාසය

ඔබට වැඩිපුරම පහසු user interface එක කුමක්ද? හේතුව ලියන්න.

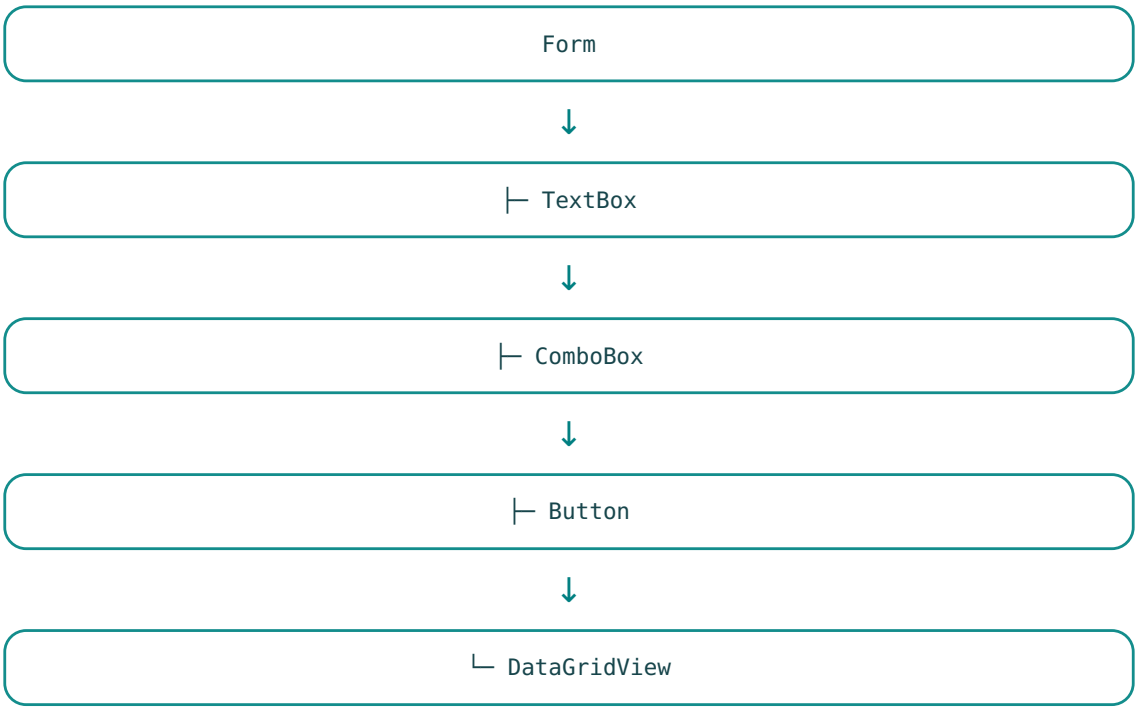
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

ඕනෑම සාධාරණ හේතුවක් සරි.

LESSON 007 / 143

Windows Forms හඳුනාගනිමු

WinForms කියන්නේ .NET භාවිතයෙන් Windows desktop windows, buttons, forms, tables නිර්මාණය කරන UI technology එකකි. මෙය Windows සඳහා පමණක් ක්‍රියාත්මක වෙයි.



ඇයි මෙය වැදගත්?

අපේ app එක offline desktop learning project එකක්; cloud service එකක් හෝ bank application එකක් නොවේ.

පුංචි අභ්‍යාසය

DataGridView එකක පෙන්වීමට සුදුසු දත්ත දෙකක් ලියන්න.

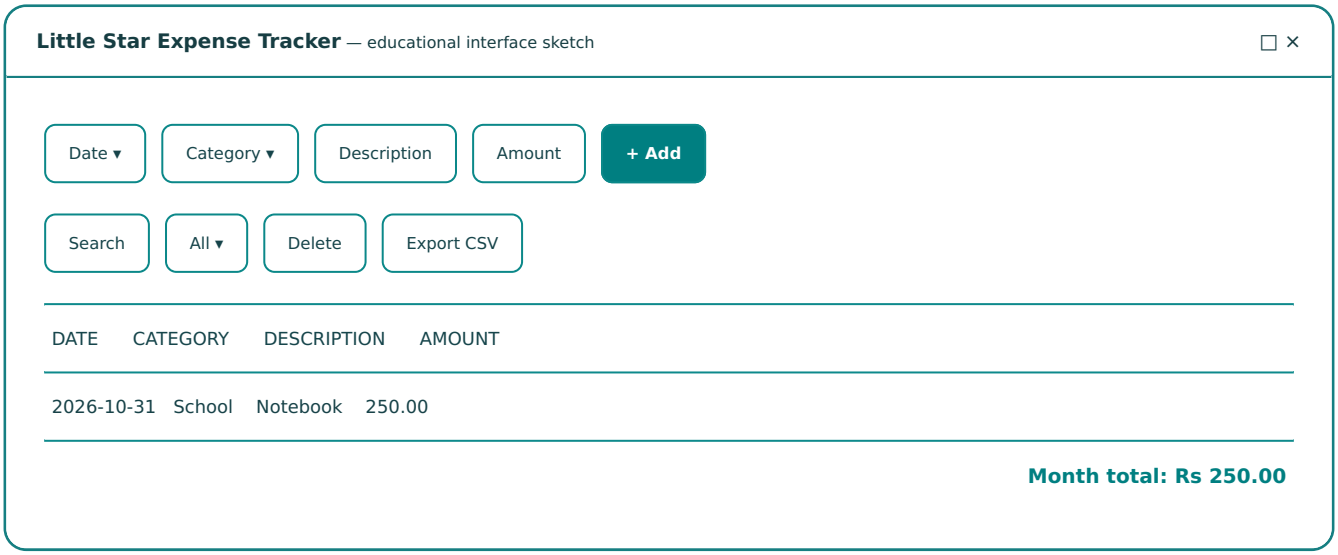
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

උදාහරණ: දිනය, වර්ගය, මුදල.

LESSON 008 / 143

කියවන්න කලින් අඳින්න

Software හදන ගමන coding වලින් පමණක් පටන් නොගනී. ප්‍රශ්නය දැනගෙන කඩදාසියක sketch එකක් අඳින එකත් හොඳ පළමු පියවරක්.



ඇයි මෙය වැදගත්?

UI sketch එකක් අවසාන design එක නොවේ. වෙනස් කරන්න ලේසියි; එවිට කාලය ඉතිරි වෙනවා.

පුංචි අභ්‍යාසය

කඩදාසියක ඔබේ app එකේ Add button එක තැබිය යුතු තැන අඳින්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Form එකේ ඉහළින් input fields අසල තබන්න පුළුවන්.

LESSON 009 / 143

අපේ Example Project එක

Little Star Expense Tracker නම් කල්පිත school-project එක සාදනවා. Expense කියන්නේ වියදමක්; විස්තරය, දිනය, වර්ගය සහ amount අපි ගබඩා කරනවා.

Windows Command Prompt / CMD

```
Food   → Rs 120.00
School → Rs 250.00
Travel → Rs  90.00
Total  → Rs 460.00
```

ඇයි මෙය වැදගත්?

මෙය real financial records සඳහා නොව ඉගෙනීමට පමණයි. කල්පිත දත්ත පමණක් ඇතුළත් කරන්න.

පුංචි අභ්‍යාසය

මෙම records තුනේ එකතුව නැවත ගණනය කර බලන්න.

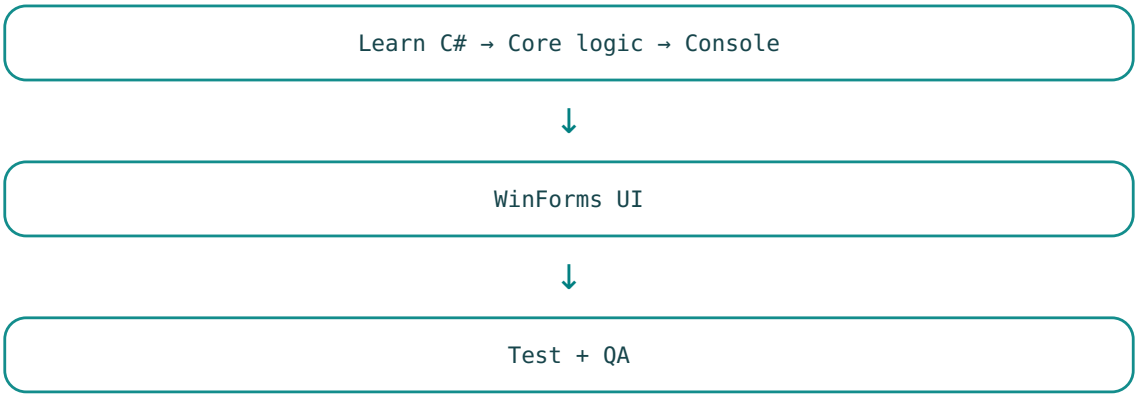
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

120 + 250 + 90 = 460.

LESSON 010 / 143

අපේ ගමනේ සිතියම

මුලින් simple console program එක, පසුව data rules, JSON storage, WinForms buttons සහ අවසානයේ tests ලියනවා. මේ විදිහට පියවරෙන් පියවර වැඩ කළ විට දරුවෙකුටත් සම්පූර්ණ app එකක් තේරුම් ගන්න පුළුවන්.



ඇයි මෙය වැදගත්?

ලොකු project එකක් පොඩි කොටස්වලට කඩා සැලසුම් කිරීම software engineering හි වැදගත් පුරුද්දකි.

පුංචි අභ්‍යාසය

ඔබට අපේ ගමනේ වැඩිපුරම කැමති කොටස කුමක්ද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

කැමති කොටසක් තෝරලා හේතුව ලියන්න.

LESSON 011 / 143

Windows සහ Install සැලැස්ම

මෙම පොතේ Windows Forms app එක සඳහා Windows පරිගණකයක් අවශ්‍යයි. Visual Studio 2026 Community සහ .NET 10 SDK එක setup කරන්න. Installer එකේ `.NET desktop development` workload එක තෝරන්න.

```
Microsoft official links:
        ↓
https://visualstudio.microsoft.com/
        ↓
https://dotnet.microsoft.com/download
```

ඇයි මෙය වැදගත්?

ස්ථාපනය කරනවිට Microsoft නිල sites පමණක් භාවිතා කරන්න. අවශ්‍යනම් වැඩිහිටි අයෙකුගෙන් උදව් ගන්න.

පුංචි අභ්‍යාසය

Install කරගන්නා tools දෙකේ නම් ලියන්න.

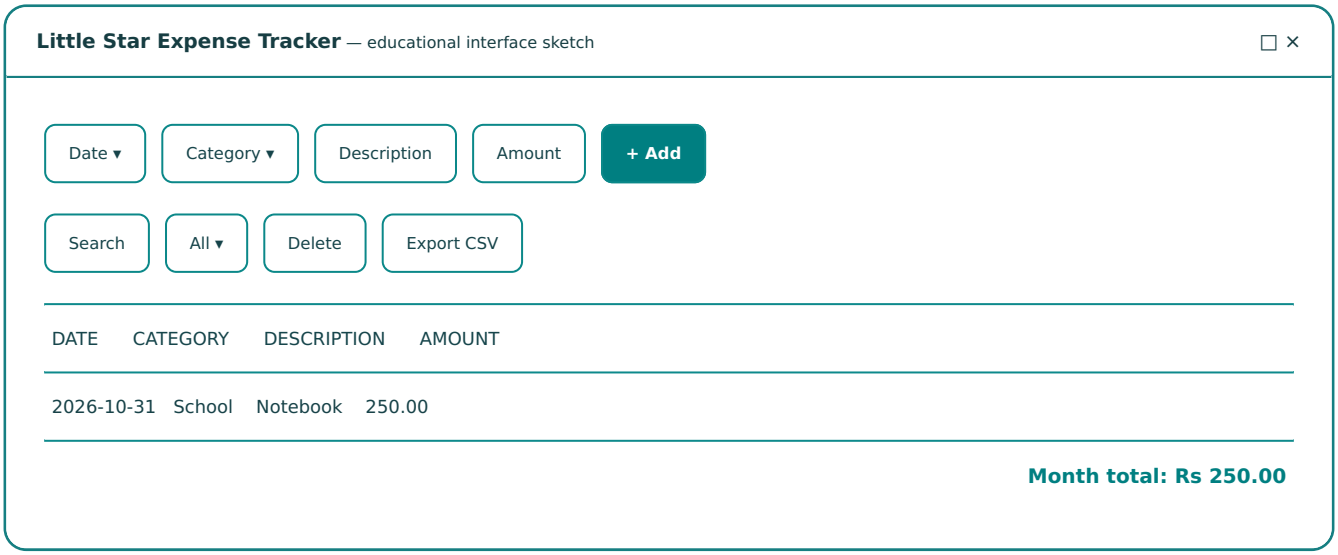
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Visual Studio 2026; .NET 10 SDK.

LESSON 012 / 143

Visual Studio 2026 ආරම්භය

Visual Studio කියන්නේ විශාල IDE එකක්. Solution Explorer හි files, editor හි code, Error List හි errors, Output හි build messages බලන්න පුළුවන්.



ඇයි මෙය වැදගත්?

VS Code වෙනුවට Visual Studio ද භාවිතා කළ හැකියි; WinForms designer සඳහා Visual Studio පහසුයි.

පුංචි අභ්‍යාසය

IDE කියන වචනයේ තේරුම සොයන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Integrated Development Environment.

LESSON 013 / 143

CMD එක විවෘත කරමු

Windows Search එකේ `cmd` කියලා type කර Command Prompt එක open කරන්න. ඉන්පසු `dotnet --version` දී SDK install වී තිබෙනවාද බලන්න.

Windows Command Prompt / CMD

```
C:\> dotnet --version
10.0.xxx
C:\>
```

ඇයි මෙය වැදගත්?

Version number එක නොපෙනේ නම් install නැති හෝ PATH සැකසුම් ගැටලුවක් විය හැකියි.

පුංචි අභ්‍යාසය

ඔබේ පරිගණකයේ `dotnet --version` run කර ලැබුණු අගය සටහන් කරන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

අංකය `10.0...` ආකාරයේ විය හැකියි.

LESSON 014 / 143

Folder එකක් සකස් කරමු

Projects තැබීමට `Documents` යටතේ `DotNetLessons` නමින් folder එකක් සාදාගන්න. CMD එකේ `cd` command එකෙන් ඒ folder එකට යන්න.

Windows Command Prompt / CMD

```
mkdir DotNetLessons
cd DotNetLessons
mkdir HelloKid
cd HelloKid
```

ඇයි මෙය වැදගත්?

එක project එකේ files තවත් project එකක files සමඟ මිශ්‍ර නොවීමට folders වැදගත්.

පුංචි අභ්‍යාසය

CMD හි `cd ..` command එකෙන් සිදු වන දේ කියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

ඉහළ parent folder එකට යනවා.

LESSON 015 / 143

පළමු Console Project එක

`dotnet new console -n HelloKid` command එකෙන් කුඩා C# project එකක් සාදනවා. `HelloKid` folder එකට ගිය විට `Program.cs` සහ `.csproj` files පෙනෙයි.

Windows Command Prompt / CMD

```
dotnet new console -n HelloKid
cd HelloKid
dotnet run
```

ඇයි මෙය වැදගත්?

.csproj file එකේ app එක build කිරීමට අවශ්‍ය project settings තියෙනවා.

පුංචි අභ්‍යාසය

මෙම commands තුන නිවැරදි අනුපිළිවෙලින් ලියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

new → cd → run.

LESSON 016 / 143

Hello World හදමු

`Program.cs` file එක open කර `Console.WriteLine` භාවිතා කරන්න. එයින් CMD window එකට පණිවුඩයක් පෙන්වනවා. මෙය අපේ පළමු C# victory එක.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
Console.WriteLine("Hello, world!");
Console.WriteLine("I am learning .NET!");
```

ඇයි මෙය වැදගත්?

C# හි spelling සහ capital letters නිවැරදිව තබන්න; `console` සහ `Console` එකම නොවේ.

පුංචි අභ්‍යාසය

දෙවන line එකේ ඔබ කැමති පණිවුඩයක් දැමී run කරන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

CMD හි lines දෙකම පෙන්විය යුතුයි.

LESSON 017 / 143

Terminal output කියවන්න

`dotnet run` කළ විට messages terminal එකේ පෙනෙනවා. එම output යනු app එක ලබාදෙන ප්‍රතිඵලයයි. Terminal එකේ command සහ output දෙක වෙන් කර හඳුනාගන්න.

Windows Command Prompt / CMD

```
C:\DotNetLessons\HelloKid> dotnet run
Hello, world!
I am learning .NET!
```

ඇයි මෙය වැදගත්?

Developer කෙනෙකුට log කියවන්න පුළුවන් වීම debugging සඳහා ඉතාමත් වටිනවා.

පුංචි අභ්‍යාසය

`dotnet build` සහ `dotnet run` අතර වෙනස ලියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

build compiles; run builds/starts the app.

LESSON 018 / 143

Source files පිළිවෙල

Console project එකේ `.csproj` configuration එකත් `Program.cs` code එකත් වෙන වෙනම තියෙනවා. `bin` සහ `obj` folders build කළ පසු ස්වයංක්‍රීයව සැදෙනවා.

PROJECT EXPLORER

```
HelloKid/
--- HelloKid.csproj
--- Program.cs
--- bin/  (generated)
--- obj/  (generated)
```

ඇයි මෙය වැදගත්?

Auto-generated files අතින් edit නොකර source files මත වැඩ කරන්න.

පුංචි අභ්‍යාසය

`Program.cs` file එකේ කාර්යය වචන දෙකකින් කියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

C# instructions.

LESSON 019 / 143

Code comments

`//` භාවිතයෙන් එක-line comment ලියන්න පුළුවන්. Comments යනු computer එක execute නොකරන, මිනිස්සුන්ට උදව් කරන සටහන්.



ඇයි මෙය වැදගත්?

අපේ source ZIP එකේ comments ඉංග්‍රීසියෙන් තබා තිබෙනවා; PDF එකේ සිංහලෙන් පැහැදිලි කරනවා.

පුංචි අභ්‍යාසය

ඔබේ code එකේ comment එකක් ලියා හේතුව පැහැදිලි කරන්න.

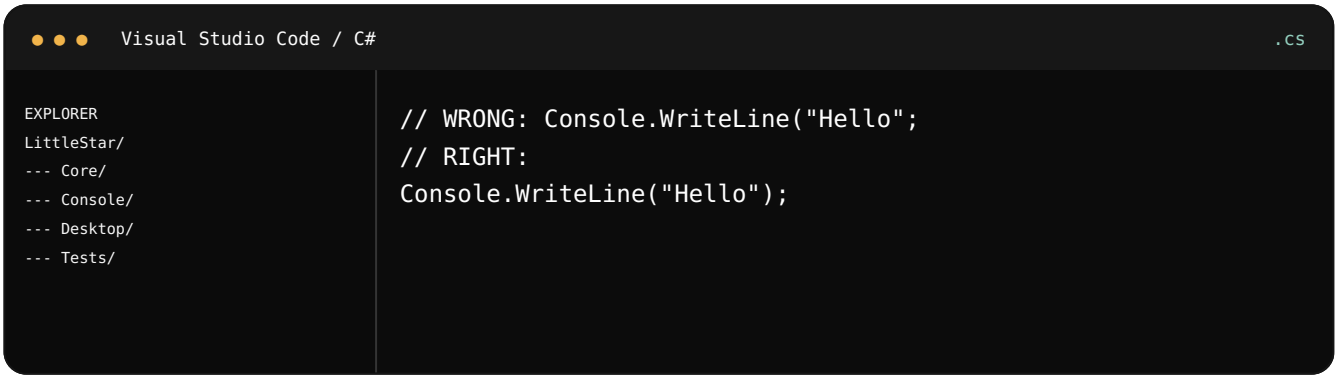
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Comment code එක ගැන සටහනක් පමණයි.

LESSON 020 / 143

පළමු error එක හදා බලමු

Code එකේ closing `)` එක මඟහැරලා `dotnet build` කරන්න. Compiler එක error එකක් පෙන්වනවා. නැවත bracket එක දාලා build කළාම error එක විසඳෙනවා.



ඇයි මෙය වැදගත්?

වැරදි හඳුනාගෙන නිවැරදි කිරීම coding ගමනේ සාමාන්‍ය කොටසකි.

පුංචි අභ්‍යාසය

Compile error එකක් දුටු විට පළමුව බැලිය යුතු තැන කුමක්ද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Error List / terminal message සහ line number.

LESSON 021 / 143

Statement එකක්

Statement කියන්නේ app එකට කරන අණක්. C# හි බොහෝ statements අවසානයේ `;` යොදනවා. වැරදි punctuation එකක් නිසා compiler එක අසමත් විය හැකියි.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
Console.WriteLine("A");
Console.WriteLine("B");
```

ඇයි මෙය වැදගත්?

නිවැරදි syntax ලිවීම coding හි පළමු දක්ෂතාව.

පුංචි අභ්‍යාසය

A, B, C එකිනෙක වෙනම lines වල පෙන්වන statements 3ක් ලියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Console.WriteLine("C"); ඇතුළු lines 3ක්.

LESSON 022 / 143

Variable කියන පෙට්ටිය

Variable කියන්නේ මතකයේ value එකක් තබන නමක්. `int pens = 3;` යනු `pens` නමින් 3 සුරකින එකයි.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
int pens = 3;
pens = 5;
Console.WriteLine(pens);
```

ඇයි මෙය වැදගත්?

කවදා හෝ data එක වෙනස් කිරීමට variables ඉතාමත් ප්‍රයෝජනවත්.

පුංචි අභ්‍යාසය

මෙහි output එක කීයද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

5.

LESSON 023 / 143

String සමඟ වචන

`string` කියන්නේ text සඳහා දත්ත වර්ගයක්. වචන ද්විත්ව quotation marks තුළ ලියනවා. පාරිභෝගික නම වගේ texts අගයක් ලෙස තබාගන්න පුළුවන්.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
string item = "Notebook";
Console.WriteLine(item);
```

ඇයි මෙය වැදගත්?

ගණන් හා text දෙක වෙනස් types. ඒවා වෙන් තබාගැනීම errors අඩු කරනවා.

පුංචි අභ්‍යාසය

ඔබ කැමති පොතක නම string එකක තබන්න.

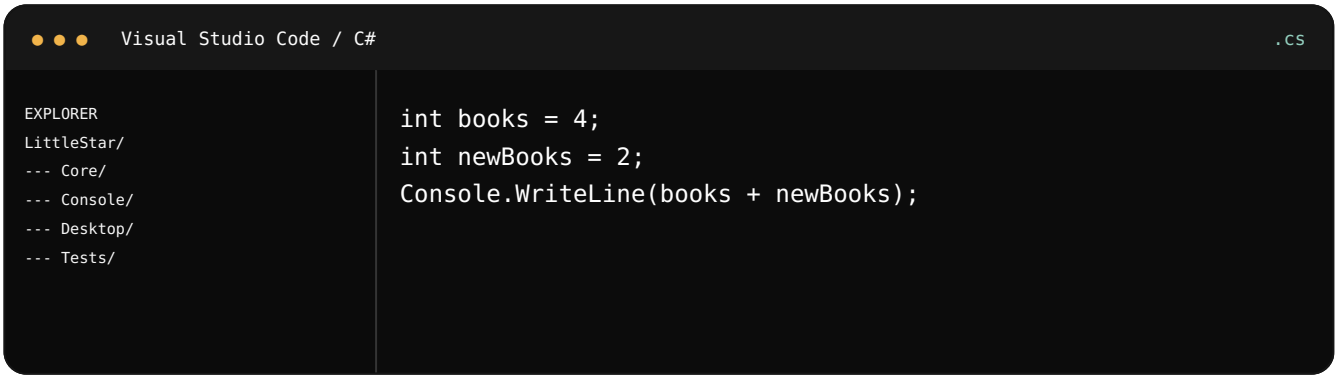
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

`string book = "...";` යන ආකාරය.

LESSON 024 / 143

int සහ සම්පූර්ණ ගණන්

`int` කියන්නේ decimal නැති සම්පූර්ණ ගණන්. ගණන, වයස, ක්‍රීඩා ලකුණු වගේ දේවල් එයින් තැබිය හැකියි.



ඇයි මෙය වැදගත්?

Rs 25.50 වගේ මුදල් සඳහා int නොව decimal භාවිතා කළ යුතුයි.

පුංචි අභ්‍යාසය

Output එක කීයද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

6.

LESSON 025 / 143

decimal සහ මුදල්

C# හි මුදල් ගණනයට `decimal` ප්‍රයෝජනවත්. `m` suffix එකක් දීමෙන් decimal literal එකක් බව පෙන්වනවා.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
decimal pen = 75.50m;
decimal book = 250.00m;
Console.WriteLine(pen + book);
```

ඇයි මෙය වැදගත්?

`float` හා `double` binary approximations විය හැකි නිසා මුදල් ගණන් පිළිබඳ උදාහරණයට decimal තෝරනවා.

පුංචි අභ්‍යාසය

`125.25m` අගයක් දාලා totals ගණනය කරන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

75.50 + 250 = 325.50.

LESSON 026 / 143

bool සහ true / false

`bool` වල අගයන් දෙකයි: `true` හෝ `false`. App එකේ පරීක්ෂා කිරීමට මෙය උපකාරී.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
bool isValid = true;
Console.WriteLine(isValid);
```

ඇයි මෙය වැදගත්?

User විසින් නිවැරදිව form එක පිරවුවාද කියා check කිරීමට conditions අවශ්‍ය වෙනවා.

පුංචි අභ්‍යාසය

`isOpen` කියන bool variable එක false ලෙස ලියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

bool isOpen = false;

LESSON 027 / 143

User input කියවමු

Console app එකේ `Console.ReadLine()` භාවිතයෙන් keyboard එකෙන් text ලබාගන්න පුළුවන්. Result එක string හෝ null විය හැකියි.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
Console.Write("Your name: ");
string name = Console.ReadLine() ?? "Friend";
Console.WriteLine("Hello " + name);
```

ඇයි මෙය වැදගත්?

User input සෑමවිටම නිවැරදි කියා විශ්වාස නොකරන්න.

පුංචි අභ්‍යාසය

නම ඇතුළත් කළ පසු personalized greeting එකක් පෙන්වන app එකක් හදන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Hello කියලා නම පෙන්වෙයි.

LESSON 028 / 143

ඉලක්කම් text එකෙන් number එකට

CMD එකෙන් ලැබෙන `"250"` යනු string එකක්. ගණනයට `decimal.TryParse` යොදා number එකක් බවට පරිවර්තනය කරන්න.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
bool ok = decimal.TryParse("250.50", out decimal value);
Console.WriteLine(ok);
Console.WriteLine(value);
```

ඇයි මෙය වැදගත්?

`TryParse` වැරදි user input එකක් ලැබුණාම crash නොවී check කිරීමට උපකාරී.

පුංචි අභ්‍යාසය

`"abc"` දුන්නොත් `ok` true ද false ද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

false.

LESSON 029 / 143

Math operators

`+`, `-`, `*`, `/` වැනි operators මඟින් ගණනය කරනවා. අපේ expense total එකට එකතු කිරීම වැදගත්.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
decimal x = 100m;
decimal y = 40m;
Console.WriteLine(x - y);
```

ඇයි මෙය වැදගත්?

Data type එක අනුව division result වෙනස් විය හැකියි.

පුංචි අභ්‍යාසය

x + y, x * y, x / y වල values ගණනය කරන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

140, 4000, 2.5.

LESSON 030 / 143

Comparison operators

`==`, `!=`, `>`, `<` වැනි සංකේත value දෙකක් සසඳනවා. ප්‍රතිඵලය `true` හෝ `false`.

Visual Studio Code / C# .cs

EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
decimal amount = 200m;
Console.WriteLine(amount > 0m);
Console.WriteLine(amount == 100m);
```

ඇයි මෙය වැදගත්?

අපේ app එකේ amount positive ද කියලා මෙයින් පරීක්ෂා කරනවා.

පුංචි අභ්‍යාසය

Output lines දෙකේ true/false ලියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

true, false.

LESSON 031 / 143

if / else තීරණයක්

`if` condition එක true නම් එක් code block එකක්, නැත්නම් `else` එකක් execute කරනවා.

Visual Studio Code / C# .cs

EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
decimal amount = -5m;
if (amount > 0)
    Console.WriteLine("OK");
else
    Console.WriteLine("Invalid");
```

ඇයි මෙය වැදගත්?

Invalid data save කිරීම වැළැක්වීමට if/else වැදගත්.

පුංචි අභ්‍යාසය

amount = 50m කළොත් output එක කුමක්ද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

OK.

LESSON 032 / 143

AND සහ OR

`&&` යනු AND, `||` යනු OR. Condition දෙකම හෝ එකක් හෝ සත්‍ය විය යුතුද කියා තීරණය කරනවා.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
bool amountOk = true;
bool categoryOk = false;
Console.WriteLine(amountOk && categoryOk);
```

ඇයි මෙය වැදගත්?

Amount valid සහ category valid දෙකම පරීක්ෂා කරනවා.

පුංචි අභ්‍යාසය

AND ප්‍රතිඵලය false වෙන්නේ ඇයි?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

එක condition එකක් false නිසා.

LESSON 033 / 143

switch තේරීම්

Menu එකක 1, 2, 3 යන තේරීම් වලට වෙනම actions දිය හැකියි. `switch` එකෙන් ඒ logic එක පැහැදිලි කරගන්න පුළුවන්.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
string choice = "2";
switch (choice)
{
    case "1": Console.WriteLine("Add"); break;
    case "2": Console.WriteLine("List"); break;
}
```

ඇයි මෙය වැදගත්?

CMD menu එක readable කිරීමට switch/else-if යොදාගන්න පුළුවන්.

පුංචි අභ්‍යාසය

choice 1 කළොත් output එක කුමක්ද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Add.

LESSON 034 / 143

for loop

`for` කියන්නේ එක ක්‍රියාවක් නියමිත වාර ගණනක් නැවත කරන loop එකක්.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
for (int i = 1; i <= 5; i++)
{
    Console.WriteLine(i);
}
```

ඇයි මෙය වැදගත්?

5 වතාවක් item number පෙන්වීම වගේ tasks සඳහා.

පුංචි අභ්‍යාසය

i <= 3 කළොත් output එක කුමක්ද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

1, 2, 3.

LESSON 035 / 143

while loop

`while` condition එක true තිබෙන තාක් නැවත run වෙනවා. User `0` තෝරන තුරු console menu එක පෙන්වන්න මේක සුදුසුයි.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
int count = 1;
while (count <= 3)
{
    Console.WriteLine(count);
    count++;
}
```

ඇයි මෙය වැදගත්?

Loop එකෙන් පිටවෙන condition එක තිබීම වැදගත්; නැත්නම් infinite loop ඇතිවිය හැකියි.

පුංචි අභ්‍යාසය

count++ මඟින් කුමක් සිදුවේද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

count එක 1කින් වැඩි වෙනවා.

LESSON 036 / 143

foreach loop

Collection එකක තිබෙන හැම item එකක්ම සරලව කියවීමට `foreach` භාවිතා කරනවා.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
string[] items = ["Pen", "Book", "Bag"];
foreach (string item in items)
    Console.WriteLine(item);
```

ඇයි මෙය වැදගත්?

Expense list එකක records එකින් එක console එකේ පෙන්වීම සඳහා.

පුංචි අභ්‍යාසය

මෙහි output lines ගණන කීයද?

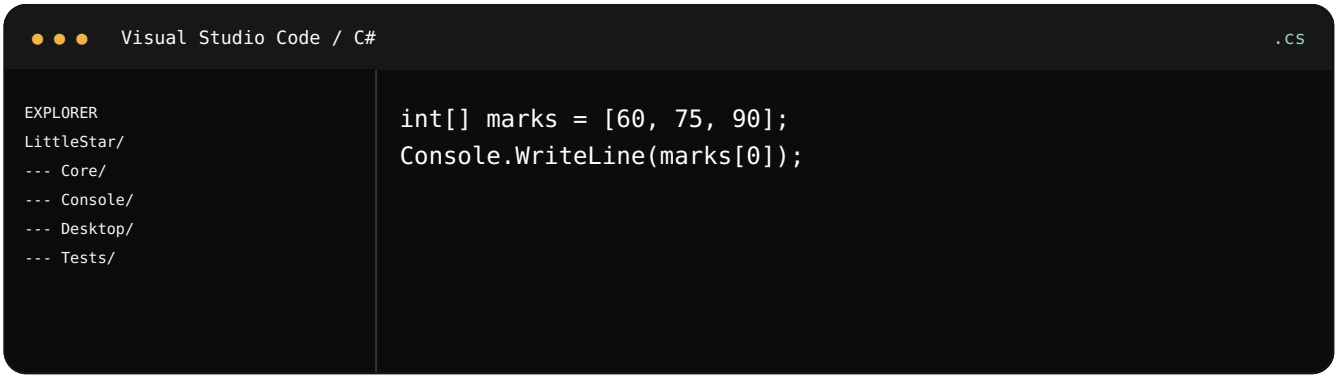
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

3.

LESSON 037 / 143

Arrays

Array එකක් එකම type එකේ values කිහිපයක් තබාගන්නවා. Elements index එකෙන් ලබාගන්නවා. පළමු index එක 0යි.



ඇයි මෙය වැදගත්?

Array fixed-size. වැඩි හෝ අඩු වෙමින් පවතින records සඳහා List වඩා පහසුයි.

පුංචි අභ්‍යාසය

marks[2] දුන්නොත් output එක?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

90.

LESSON 038 / 143

List<T> සමඟ records

`List<T>` කියන්නේ items එකතු කිරීම සහ මකා දැමීම පහසු collection එකක්. App එකේ expenses සුරකින විට මෙය භාවිතා කරනවා.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
List<string> names = ["Nila", "Amal"];
names.Add("Ravi");
Console.WriteLine(names.Count);
```

ඇයි මෙය වැදගත්?

`<T>` යනු එහි තබන type එක. List<string> කියන්නේ වචන ලැයිස්තුවක්.

පුංචි අභ්‍යාසය

`names.Count` එකේ අගය කීයද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

3.

LESSON 039 / 143

Method කියන්නේ කුමක්ද?

Method කියන්නේ නැවත භාවිතා කළ හැකි code block එකක්. එකම message එක දහවරක් ලියන්න වෙනුවට method එකක් call කරන්න.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
static void Welcome()
{
    Console.WriteLine("Welcome!");
}
Welcome();
```

ඇයි මෙය වැදගත්?

Code repetition අඩු වෙනවා; test කිරීමටත් පහසුයි.

පුංචි අභ්‍යාසය

`Bye` පෙන්වන method එකක් හදන්න.

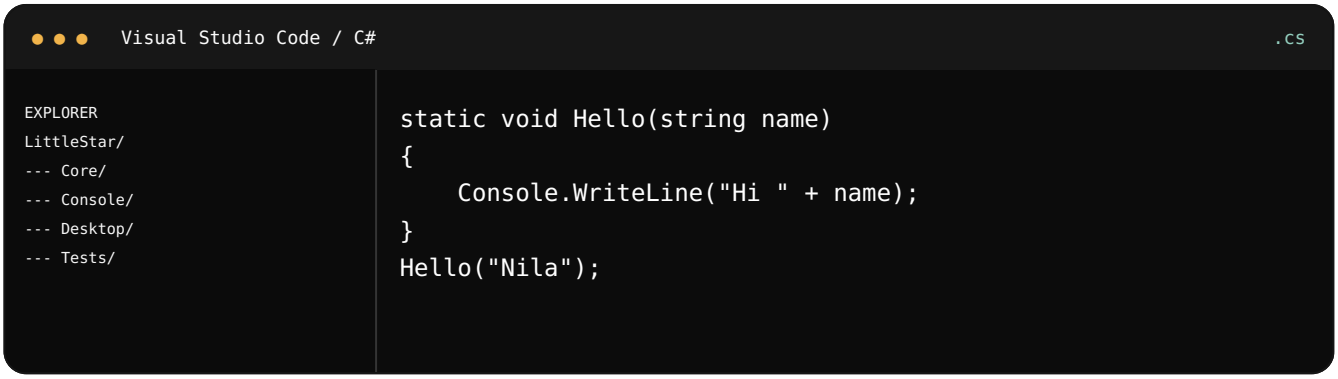
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

static void Bye() { Console.WriteLine("Bye"); }

LESSON 040 / 143

Parameters

Method එකකට value යැවීමට parameter භාවිතා කරනවා. එකම method එකට වෙනස් names යවලා වෙනස් output ලබාගන්න පුළුවන්.



ඇයි මෙය වැදගත්?

Method input සහ output පැහැදිලි වීම software design එකේ හොඳ ලක්ෂණයක්.

පුංචි අභ්‍යාසය

Hello("Amal") ලියා output එක බලන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Hi Amal.

LESSON 041 / 143

Return value

Method එකකට ගණනය කළ ප්‍රතිඵලයක් `return` කරන්න පුළුවන්. `decimal` return type එක මුදල් ගණන් සඳහා සුදුසුයි.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
static decimal Add(decimal a, decimal b)
{
    return a + b;
}
Console.WriteLine(Add(10m, 15m));
```

ඇයි මෙය වැදගත්?

Calculation logic එක buttons වලට බැඳ නොතබා method එකක තැබීම reuse සඳහා හොඳයි.

පුංචි අභ්‍යාසය

Output එක කීයද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

25.

LESSON 042 / 143

string interpolation

`$"..."` භාවිතයෙන් text සහ variables එකම string එකක පෙන්වන්න පුළුවන්.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
string item = "Book";
decimal price = 250m;
Console.WriteLine($"{item}: Rs {price:0.00}");
```

ඇයි මෙය වැදගත්?

Console reports readable කරන්න ඕනේ නම් මෙය පහසුයි.

පුංචි අභ්‍යාසය

මෙහි output එක ලියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Book: Rs 250.00.

LESSON 043 / 143

Class කියන්නේ blueprint එක

Class කියන්නේ object එකක් සාදන සැලැස්මක්. උදාහරණයට Book class එකෙන් අලුත් Book objects කිහිපයක් නිර්මාණය කළ හැකියි.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
class Student
{
    public string Name { get; set; } = "";
}
```

ඇයි මෙය වැදගත්?

විශාල application එකක data සහ actions පිළිවෙළට සකස් කිරීමට OOP හොඳ මාර්ගයක්.

පුංචි අභ්‍යාසය

Student class එකට Age property එකක් එකතු කරන්න.

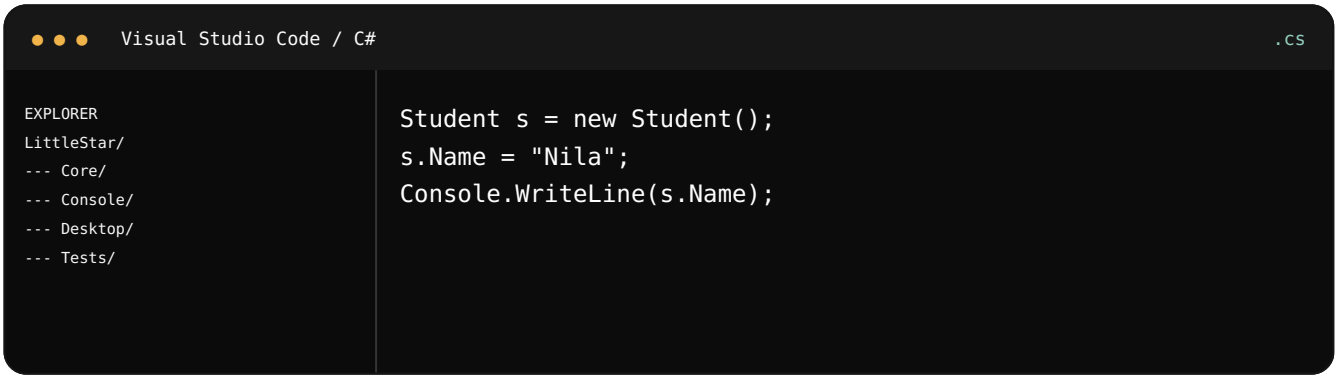
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

public int Age { get; set; }

LESSON 044 / 143

Object එකක් නිර්මාණය

Class එකේ blueprint එකෙන් object එකක් සාදා values ලබාදෙන්න පුළුවන්. `new` keyword එකෙන් අලුත් instance එකක් ලබාගන්නවා.



ඇයි මෙය වැදගත්?

App එකේ record/Expense object එකක දත්ත එකට තබනවා.

පුංචි අභ්‍යාසය

Name එක Ravi ලෙස වෙනස් කර බලන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

s.Name = "Ravi";

LESSON 045 / 143

record සමඟ data

`record` යනු data තබා ගැනීමට පහසු C# type එකක්. අපේ Expense record එකේ ID, Date, Category, Description, Amount තියෙනවා.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
public record Book(string Title, int Pages);
Book b = new("ABC", 100);
Console.WriteLine(b.Title);
```

ඇයි මෙය වැදගත්?

Record එකේ parts එකම item එකකට අදාළ වෙයි.

පුංචි අභ්‍යාසය

Book එකකට වෙනත් title එකක් දී බලන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

new("Math", 120) වගේ record එකක්.

LESSON 046 / 143

Constructor සහ initialization

Object එකක් නිර්මාණය කරන වෙලාවේ අවශ්‍ය values ලබාදීම constructor හරහා කළ හැකියි.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
class Pet
{
    public string Name;
    public Pet(string name) { Name = name; }
}
Pet p = new("Kitty");
```

ඇයි මෙය වැදගත්?

අසම්පූර්ණ object සෑදීම අඩු කරන්න constructor භාවිතා කරන්න පුළුවන්.

පුංචි අභ්‍යාසය

Pet("Puppy") නිර්මාණය කර Name පෙන්වන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Puppy.

LESSON 047 / 143

Encapsulation හා පාලනය

Object එකේ ඇතුළත list එක හැම කෙනාටම වෙනස් කරන්න ඉඩ නොදීම වැදගත්. ExpenseManager එකේ private list එකක් සහ public read-only view එකක් භාවිතා කරනවා.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
private readonly List<Expense> _expenses = [];
public IReadOnlyList<Expense> All =>
    _expenses.AsReadOnly();
```

ඇයි මෙය වැදගත්?

Rule එක bypass කර invalid expense එකක් ඇතුළත් වීම වැළැක්වීමට මේ විදිහ හොඳයි.

පුංචි අභ්‍යාසය

ඇයි `List` එකේ reference එක කෙලින් public නොකරන්නේ?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Validation rules පසෙකලා වෙනස් කිරීම් වැළැක්වීමට.

LESSON 048 / 143

try / catch

දත්ත කියවීමේ හෝ user input වලදී error ඇතිවිය හැකියි. `try` code එක run කර `catch` දී error එක හසුරුවනවා.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
try
{
    int n = int.Parse("oops");
}
catch (FormatException)
{
    Console.WriteLine("Enter a number");
}
```

ඇයි මෙය වැදගත්?

App එක හදිසියේ crash නොවී උපකාරී message එකක් දීම වැදගත්.

පුංචි අභ්‍යාසය

මෙහි output එක කුමක්ද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Enter a number.

LESSON 049 / 143

null කියන්නේ value නැහැ

C# හි null යනු දත්තයක් නොමැති බව පෙන්වන අගයක්. `Console.ReadLine()` null විය හැකි නිසා `??` භාවිතයෙන් fallback එකක් දෙන්න.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
string? name = Console.ReadLine();
Console.WriteLine(name ?? "Friend");
```

ඇයි මෙය වැදගත්?

Null related errors අඩු කිරීමට types සහ checks වැදගත්.

පුංචි අභ්‍යාසය

`??` operator එකේ කාර්යය කියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

වම null නම් දකුණු value එක භාවිතා කරයි.

LESSON 050 / 143

DateOnly හා DateTime

Expense එකකට බොහෝවිට දිනය පමණක් අවශ්‍යයි. `DateOnly` භාවිතයෙන් date එකක්, `DateTime` භාවිතයෙන් දිනය සහ වේලාව තබාගන්න පුළුවන්.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
DateOnly today = DateOnly.FromDateTime(DateTime.Today);
Console.WriteLine(today);
```

ඇයි මෙය වැදගත්?

Data model එකට අවශ්‍ය ප්‍රමාණයේ තොරතුරු පමණක් තබන්න.

පුංචි අභ්‍යාසය

පසුගිය දවසක් DateOnly ලෙස නිර්මාණය කරන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

new DateOnly(2026, 10, 30).

LESSON 051 / 143

Namespace සහ using

Namespace කියන්නේ classes එකම කණ්ඩායමකට තැබීමේ ක්‍රමය. `using` කියන්නේ වෙනත් namespace එකේ types කෙටියෙන් භාවිතා කිරීමට උපකාරී.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
using System;
namespace LittleStar.Demo;
class Hello { }
```

ඇයි මෙය වැදගත්?

Project එක විශාල වුණත් code අවුල් නොවීමට namespace වැදගත්.

පුංචි අභ්‍යාසය

Namespace එකකට කැමති නමක් දී class එකක් ලියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

namespace MySchool; class Pupil {}

LESSON 052 / 143

Compiler warning සහ error

Error එකක් තිබුණොත් build නවතී. Warning එකක් නම් build හැකි වුවත් potential problem එකක් පෙන්වනවා.

Windows Command Prompt / CMD

```
dotnet build
// Read terminal messages:
// Error: must fix
// Warning: inspect and improve
```

ඇයි මෙය වැදගත්?

Warnings නොසලකා හැරීම වෙනුවට ඒවායේ අර්ථය තේරුම් ගන්න.

පුංචි අභ්‍යාසය

ඉතා වැදගත් rule එක: error දුටු විට මුල් error එකෙන්ම පටන්ගන්න.

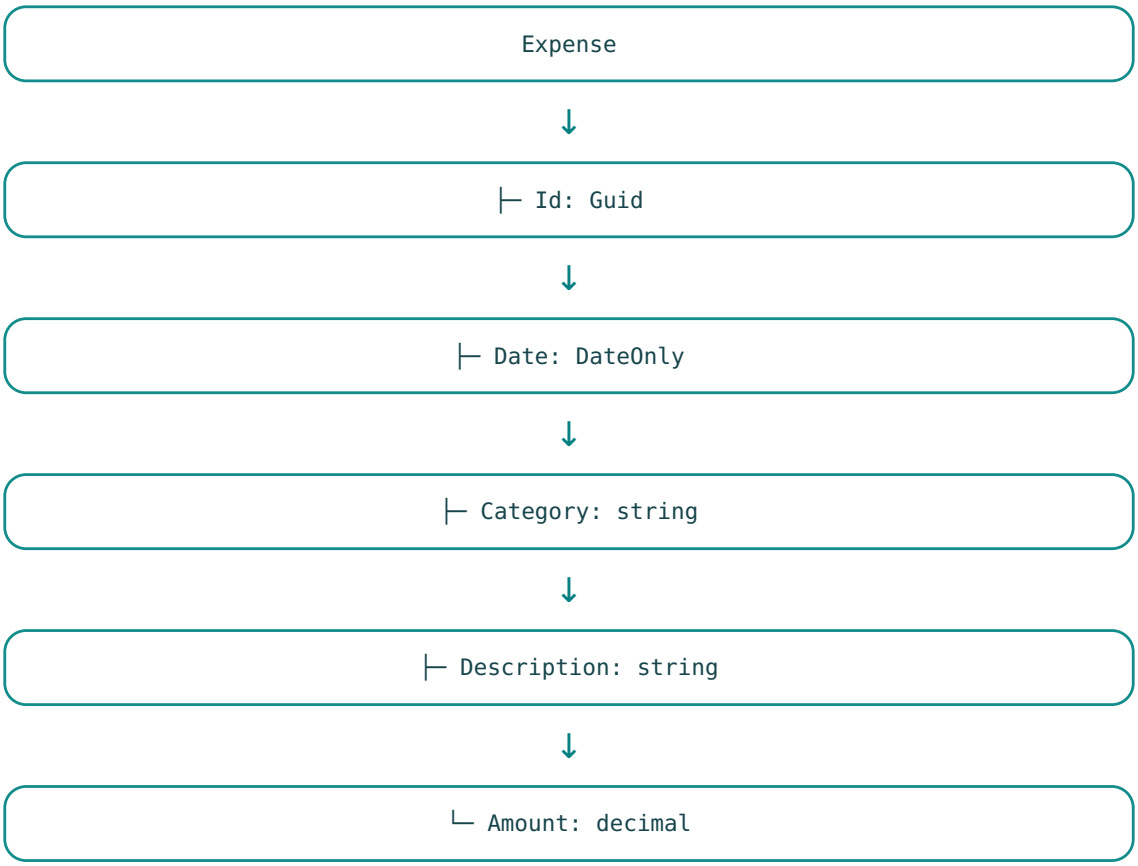
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

පළමු error එක නිවැරදි කර නැවත build කරන්න.

LESSON 053 / 143

Data model සැලසුම

Expense එකක් ගබඩා කිරීමට අවශ්‍ය fields පහක් තෝරමු: Id, Date, Category, Description, Amount. මෙය අපේ app එකේ කුඩා data model එකයි.



ඇයි මෙය වැදගත්?

පසුකාලීන වෙනස්කම් කිරීමට පෙර fields නිවැරදිව නිර්වචනය කරන්න.

පුංචි අභ්‍යාසය

Date සහ Amount සඳහා string භාවිතා නොකරන්නේ ඇයි?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

ඒවාට ගැළපෙන typed operations සුරක්ෂිතව කළ හැකි නිසා.

LESSON 054 / 143

Guid යනු unique ID

Expense records දෙකම එකම description එකෙන් තිබුණත් වෙන වෙනම ID තිබිය යුතුයි. `Guid.NewGuid()` මඟින් අලුත් unique identifier එකක් ලබාගන්න පුළුවන්.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
Guid id = Guid.NewGuid();
Console.WriteLine(id);
```

ඇයි මෙය වැදගත්?

Delete කරන විට description එකට පමණක් විශ්වාස නොකර Id එකෙන් නිශ්චිත record එක හඳුනාගන්න.

පුංචි අභ්‍යාසය

Guid එකක් අලුතෙන් සෑදූ විට මීට පෙර value එකට සමාන වෙයිද?

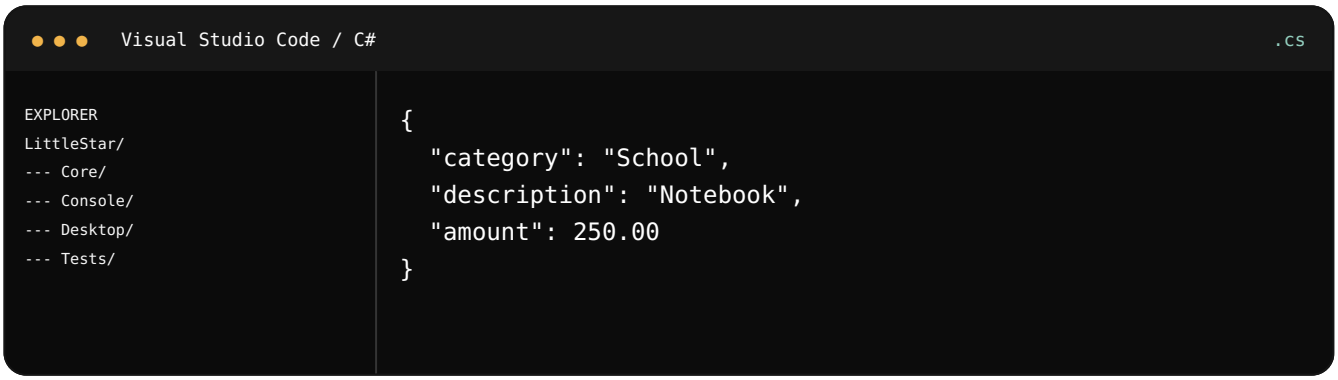
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

සාමාන්‍යයෙන් වෙනස් unique value එකක් ලැබේ.

LESSON 055 / 143

JSON ආකෘතිය

JSON යනු දත්ත text ආකාරයෙන් තබන format එකක්. `{}` object එකක්, `[]` list එකක්, key/value pairs එකින් data නිරූපණය කරනවා.



ඇයි මෙය වැදගත්?

Data file එකක් බැලීමට හැකි වීම ආරම්භකයන්ට debugging ඉගෙනීමට උපකාරී.

පුංචි අභ්‍යාසය

මෙහි keys තුන ලියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

category, description, amount.

LESSON 056 / 143

Serialization

C# objects JSON text බවට පරිවර්තනය කිරීම serialization කියනවා. .NET හි `System.Text.Json` library එක මේකට යොදාගන්න පුළුවන්.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
using System.Text.Json;
string json = JsonSerializer.Serialize(
    new { Item = "Pen", Price = 75 });
Console.WriteLine(json);
```

ඇයි මෙය වැදගත්?

Application එක වසා නැවත විවෘත කළත් data තබා ගැනීමට serialization උපකාරී.

පුංචි අභ්‍යාසය

JSON string එකේ Item සහ Price යන keys හඳුනාගන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Item, Price.

LESSON 057 / 143

Deserialization

JSON text එක නැවත C# object එකක් බවට හරවන එක deserialization. Data නොමැති නම් empty list එකක් භාවිතා කිරීම ආරක්ෂිතයි.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
string json = "[1,2,3]";
int[] numbers = JsonSerializer.Deserialize<int[]>(json) ?? [];
Console.WriteLine(numbers.Length);
```

ඇයි මෙය වැදගත්?

Invalid JSON file එකක් තිබේ නම් friendly error එකක් පෙන්විය යුතුයි.

පුංචි අභ්‍යාසය

Output එක කීයද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

3.

LESSON 058 / 143

File.ReadAllText

`File.ReadAllText(path)` මඟින් local file එකක text එකක් කියවන්න පුළුවන්. පළමුව file එක තියෙනවාද `File.Exists` මඟින් පරීක්ෂා කරන්න.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
if (File.Exists("note.txt"))
    Console.WriteLine(File.ReadAllText("note.txt"));
```

ඇයි මෙය වැදගත්?

File එක මකලා තිබුණොත් crash නොවීමට checks සහ try/catch යොදන්න.

පුංචි අභ්‍යාසය

File.Exists false නම් code එක කරන දේ මොකක්ද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

File එක නොකියවයි.

LESSON 059 / 143

File.WriteAllText

`File.WriteAllText` මඟින් text file එකක් නිර්මාණය හෝ overwrite කළ හැකියි. පරණ data අහිමි විය හැකි නිසා source code පරීක්ෂා කර run කරන්න.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
File.WriteAllText("hello.txt", "Hello from C#");
Console.WriteLine("File saved");
```

ඇයි මෙය වැදගත්?

විශේෂයෙන් files edit හෝ delete කිරීමට පෙර backup ගැනීම හොඳ පුරුද්දක්.

පුංචි අභ්‍යාසය

ඔබේ test folder එකේ `hello.txt` සාදා බලන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

File එකේ Hello from C# යන text තිබිය යුතුයි.

LESSON 060 / 143

LocalApplicationData folder

Windows user account එකට අදාළ local app data තැබීමට සුදුසු folder එක `LocalApplicationData`. අපේ simulator එක JSON ඒ යටතේ සුරකිනවා.

Visual Studio Code / C# .cs

EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
string folder = Environment.GetFolderPath(
    Environment.SpecialFolder.LocalApplicationData);
Console.WriteLine(folder);
```

ඇයි මෙය වැදගත්?

`Program Files` යටතේ data save නොකරන එක Windows permissions සඳහා හොඳයි.

පුංචි අභ්‍යාසය

ඔබේ Windows account එකේ folder path එක සටහන් කරන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

සාමාන්‍යයෙන් AppData\Local path එකක්.

LESSON 061 / 143

CSV කියන්නේ වගුවක්

CSV කියන්නේ comma-separated text rows. Excel හෝ වෙනත් table tools මඟින් කියවන්න පහසුයි. නමුත් description එකක comma තිබුණොත් quotation marks යොදනවා.

Windows Command Prompt / CMD

```
Date,Category,Description,Amount
2026-10-01,School,Notebook,250.00
2026-10-02,Food,"Apple, red",100.00
```

ඇයි මෙය වැදගත්?

Export කරන data එක නැවත ගොනුවකින් විවෘත කළ හැකිද කියා QA කරන්න.

පුංචි අභ්‍යාසය

දෙවැනි row එකේ quotation marks අවශ්‍ය ඇයි?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Description එක තුළ comma ඇති නිසා.

LESSON 062 / 143

Data privacy මුල් නීතිය

Local JSON file එක encryption නොමැතිව තබනවා. එනිසා මෙය financial secrecy අවශ්‍ය වැඩ සඳහා සුදුසු නැහැ. Practice සඳහා invented amounts පමණක් භාවිතා කරන්න.

```
SAFE:    School / Notebook / Rs 250
↓
AVOID:   Real bank account number
↓
AVOID:   Real card or password
```

ඇයි මෙය වැදගත්?

App එකේ limitations පිළිබඳව පැහැදිලිව කියන එක responsible software development එකකි.

පුංචි අභ්‍යාසය

මේ project එකට දාන්න නොකළ යුතු රහස් තොරතුරු දෙකක් ලියන්න.

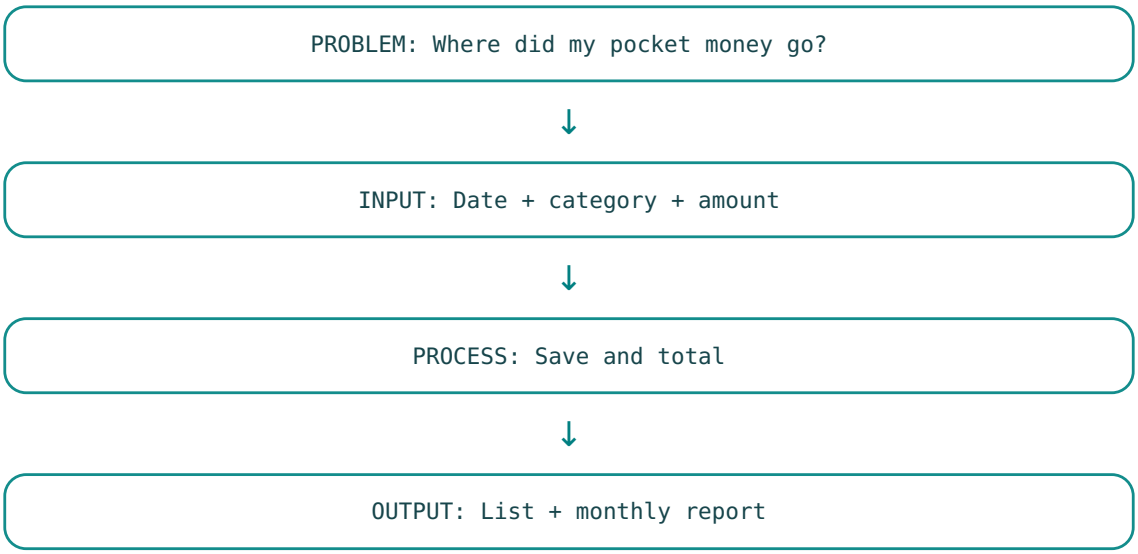
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Password සහ card number.

LESSON 063 / 143

වැඩ පටන්ගන්න කලින් ප්‍රශ්නය

User කෙනෙකුට දවසේ වියදම් ලියාගෙන මාසික එකතුව බලන්න ඕනේ. ඒ ප්‍රශ්නය වෙනුවෙන් අපේ app එක සාදනවා. ප්‍රශ්නය හරි නම් solution එක හොඳයි.



ඇයි මෙය වැදගත්?

Software කියන්නේ coding පමණක් නොව user problem එකකට ක්‍රමවත් විසඳුමකි.

පුංචි අභ්‍යාසය

මෙම app එකේ input දෙකක් සහ output දෙකක් ලියන්න.

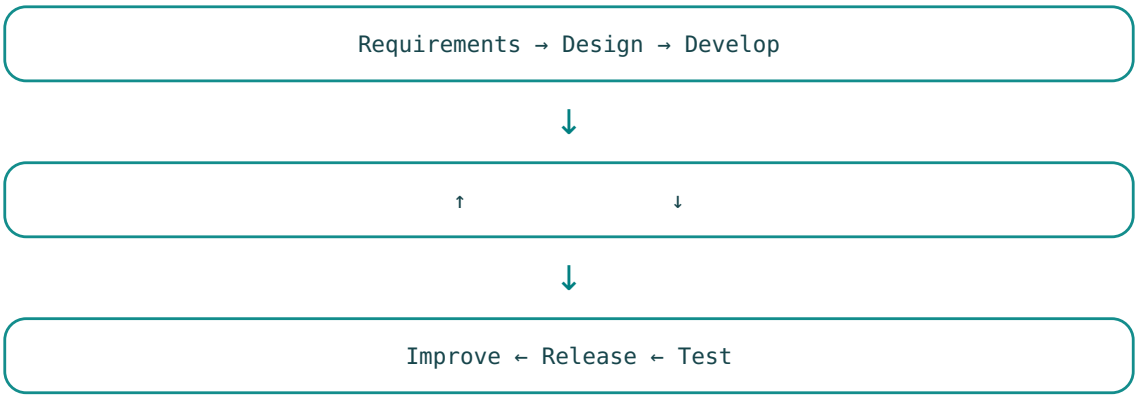
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Input: date, amount. Output: list, total.

LESSON 064 / 143

SDLC අදියර

Software Development Life Cycle (SDLC) කියන්නේ planning සිට maintenance දක්වා සම්පූර්ණ ගමන. Requirements → Design → Code → Test → Release → Improve.



ඇයි මෙය වැදගත්?

Test වාර්තාවක් හෝ user feedback එකක් නොමැතිව code ලියා ඉවර වීම project අවසානය නොවේ.

පුංචි අභ්‍යාසය

SDLC හි coding ට පෙර පියවර දෙක ලියන්න.

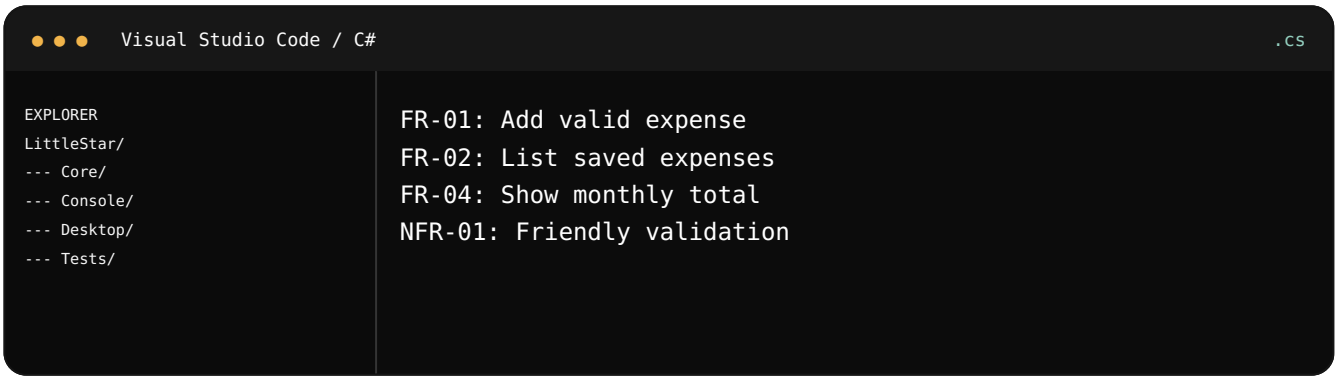
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Requirements සහ Design.

LESSON 065 / 143

SRS කියන්නේ මොකක්ද?

Software Requirements Specification (SRS) යනු app එක කළ යුතු වැඩ පැහැදිලි කරන document එකක්. Example: FR-01 Add expense, FR-04 Monthly total.



ඇයි මෙය වැදගත්?

Developer සහ tester එකම expected behavior එක ගැන එකඟවීමට SRS උපකාරී.

පුංචි අභ්‍යාසය

ඔබේම FR-08 නමින් අලුත් feature එකක් ලියන්න.

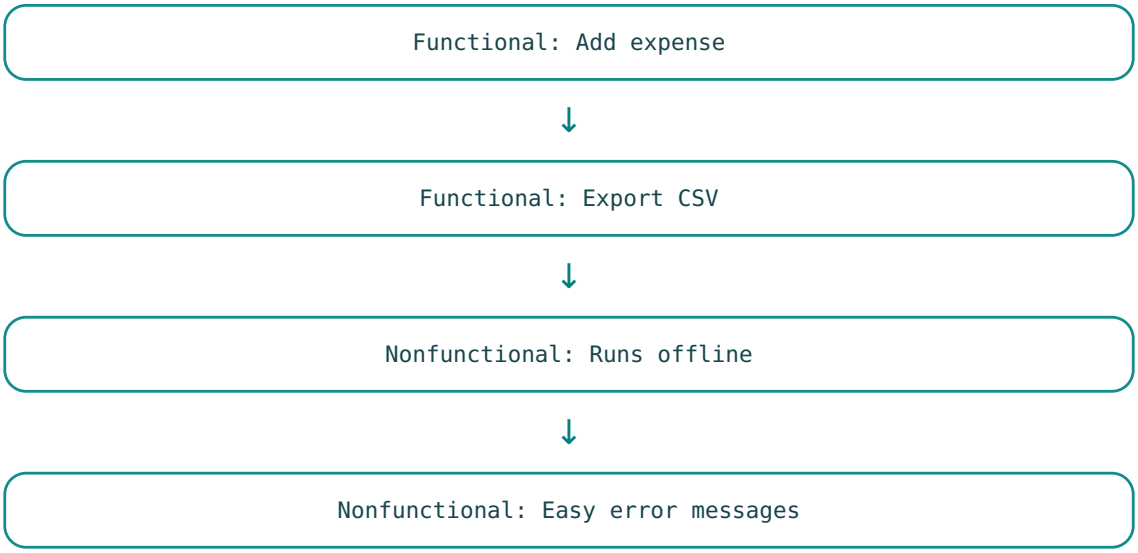
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

උදාහරණ: FR-08 Export monthly CSV.

LESSON 066 / 143

Functional vs Nonfunctional

Functional requirement යනු app කරන වැඩක්. Nonfunctional requirement යනු speed, usability, offline support වැනි ගුණාංග.



ඇයි මෙය වැදගත්?

User experience එක නිවැරදිව සැලසුම් කිරීමේදී දෙකම වැදගත්.

පුංචි අභ්‍යාසය

“App එක network නැතිව වැඩ කළ යුතුයි” යන එක කුමන requirement එකද?

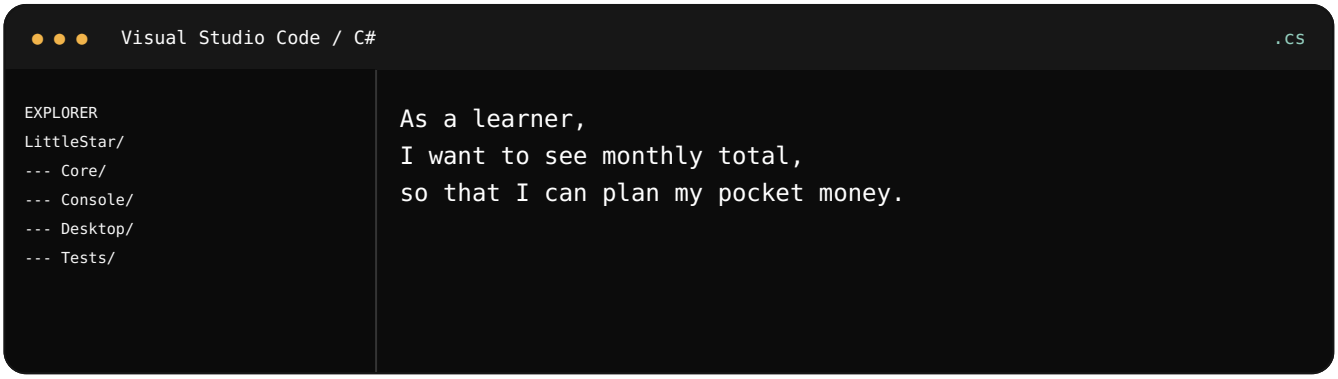
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Nonfunctional.

LESSON 067 / 143

User Story

User story එකෙන් user කෙනෙකුගේ අවශ්‍යතාව හඳුනාගන්න: “As a learner, I want to add an expense so that I can see my spending.”



ඇයි මෙය වැදගත්?

Feature එකක් නිර්මාණය කරන විට user benefit එක මතක තබාගන්න.

පුංචි අභ්‍යාසය

Delete feature එකට user story එකක් ලියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

As a learner, I want to remove a wrong entry...

LESSON 068 / 143

Acceptance Criteria

Feature එකක් අවසන්ද කියා තීරණය කිරීමට measurable rules හදන්න. Add button එක valid input save කළ යුතුයි; zero amount reject කළ යුතුයි.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
Given amount = 0
When I click Add
Then show an error
And do not save a record
```

ඇයි මෙය වැදගත්?

Test case එකකට ඒ criteria ආරම්භක point එක වේ.

පුංචි අභ්‍යාසය

Amount 100 සඳහා acceptance criterion එකක් ලියන්න.

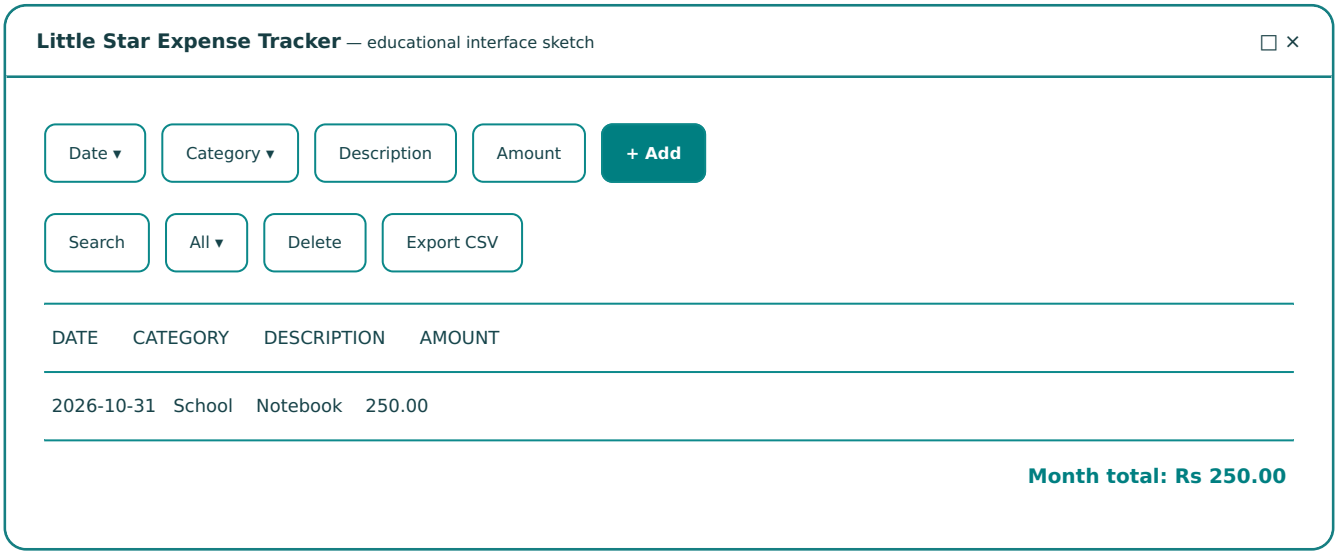
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Valid record එක save වෙලා list එකේ පෙන්විය යුතුයි.

LESSON 069 / 143

UI Wireframe

Code ලියන්න කලින් user screen එක අඳින්න. Input fields, Add button, Search box, table, Total label සරලව පෙළගස්වන්න.



ඇයි මෙය වැදගත්?

අමුතු design decoration වලට වඩා සරලව භාවිතා කළ හැකි layout එක වැදගත්.

පුංචි අභ්‍යාසය

Wireframe එකේ වැරදි input සඳහා error message තැනක් අඳින්න.

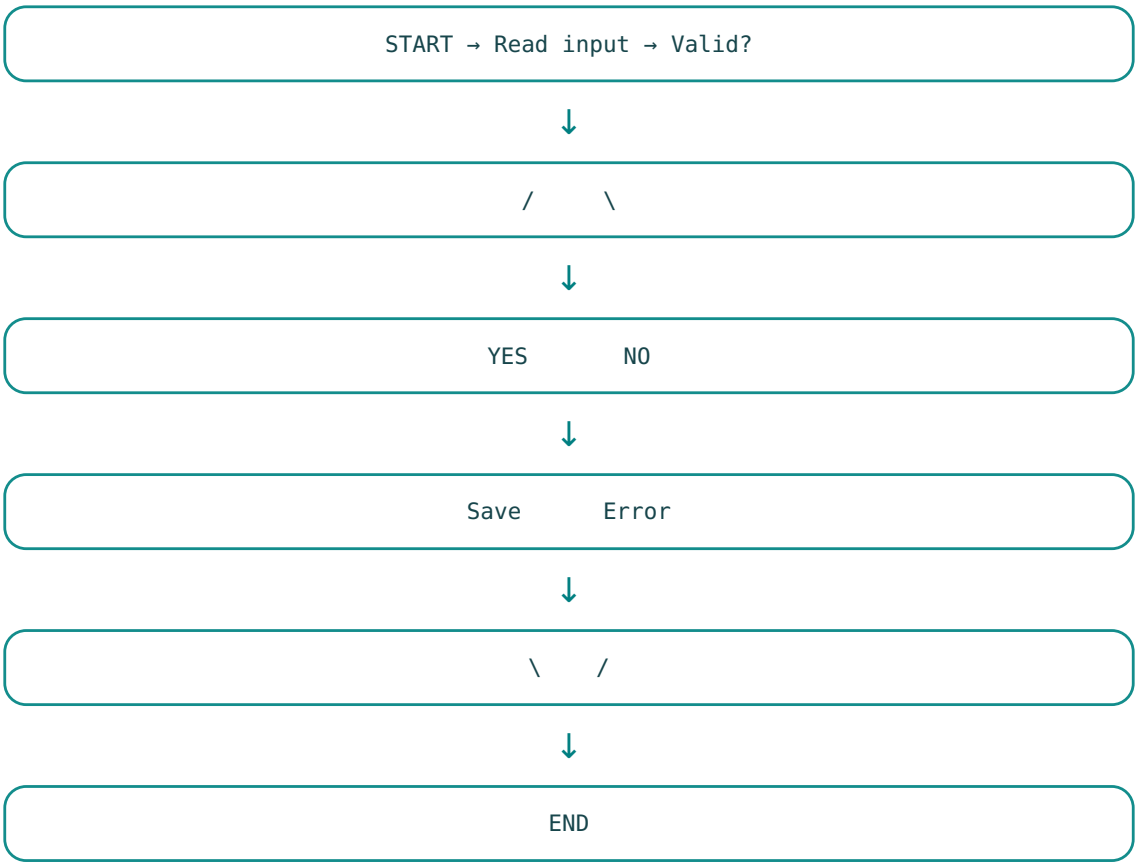
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Add input fields අසල message එකක් තැබිය හැකියි.

LESSON 070 / 143

Flowchart - Add Expense

Flowchart එකෙන් condition සහ action සම්බන්ධ වන්නේ කොහොමද කියා පෙන්වනවා. Amount valid නම් save කරන්න; නැත්නම් error පෙන්වන්න.



ඇයි මෙය වැදගත්?

Flowchart බලන tester කෙනෙකුට paths දෙකම test කළ යුතු බව වැටහෙනවා.

පුංචි අභ්‍යාසය

Invalid description එකට වෙනම decision එකක් එකතු කරන්න.

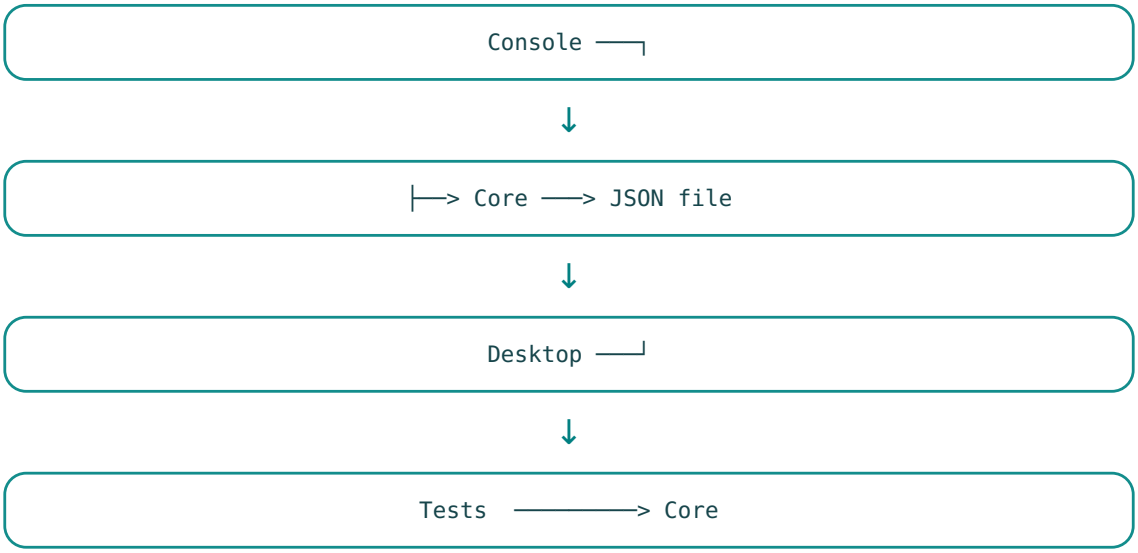
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Description not blank? → YES/NO branch.

LESSON 071 / 143

App Architecture

Core project එකේ rules. Console project එකේ menu. Desktop project එකේ buttons. Tests project එකේ automated checks. මෙය separation of concerns නම් සංකල්පයයි.



ඇයි මෙය වැදගත්?

Code එක කුඩා responsibility-based parts වලට බෙදීම maintenance පහසු කරනවා.

පුංචි අභ්‍යාසය

Console සහ Desktop දෙකම share කරන කොටස කුමක්ද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Core.

LESSON 072 / 143

Folder structure එක

ZIP එකේ folders හතර මතක තබාගන්න: Core, Console, Desktop, Tests. ඒවායේ `.csproj` files එකිනෙකට අදාළ targets කියනවා.

PROJECT EXPLORER

```
LittleStarExpenses/
--- LittleStarExpenses.Core/
--- LittleStarExpenses.Console/
--- LittleStarExpenses.Desktop/
--- LittleStarExpenses.Tests/
--- docs/
```

ඇයි මෙය වැදගත්?

Projects වෙන වෙනම තැබීමෙන් UI code සහ business logic මිශ්‍ර වන්නේ නැහැ.

පුංචි අභ්‍යාසය

ExpenseManager.cs තිබිය යුතු project එක කුමක්ද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

LittleStarExpenses.Core.

LESSON 073 / 143

File names සහ versioning

`Expense.cs`, `ExpenseManager.cs`, `ExpenseStore.cs` වැනි නම් file එකේ අරමුණ කියනවා. Git commit සමඟ version history තබාගැනීමත් හොඳ පුරුද්දක්.

PROJECT EXPLORER

```
Core/Expense.cs
Core/ExpenseManager.cs
Core/ExpenseStore.cs
Docs/SRS.md
```

ඇයි මෙය වැදගත්?

`final_new_latest2.cs` වගේ අවුල් නම් අඩු කරන්න.

පුංචි අභ්‍යාසය

Bug report template file එක කුමක්ද?

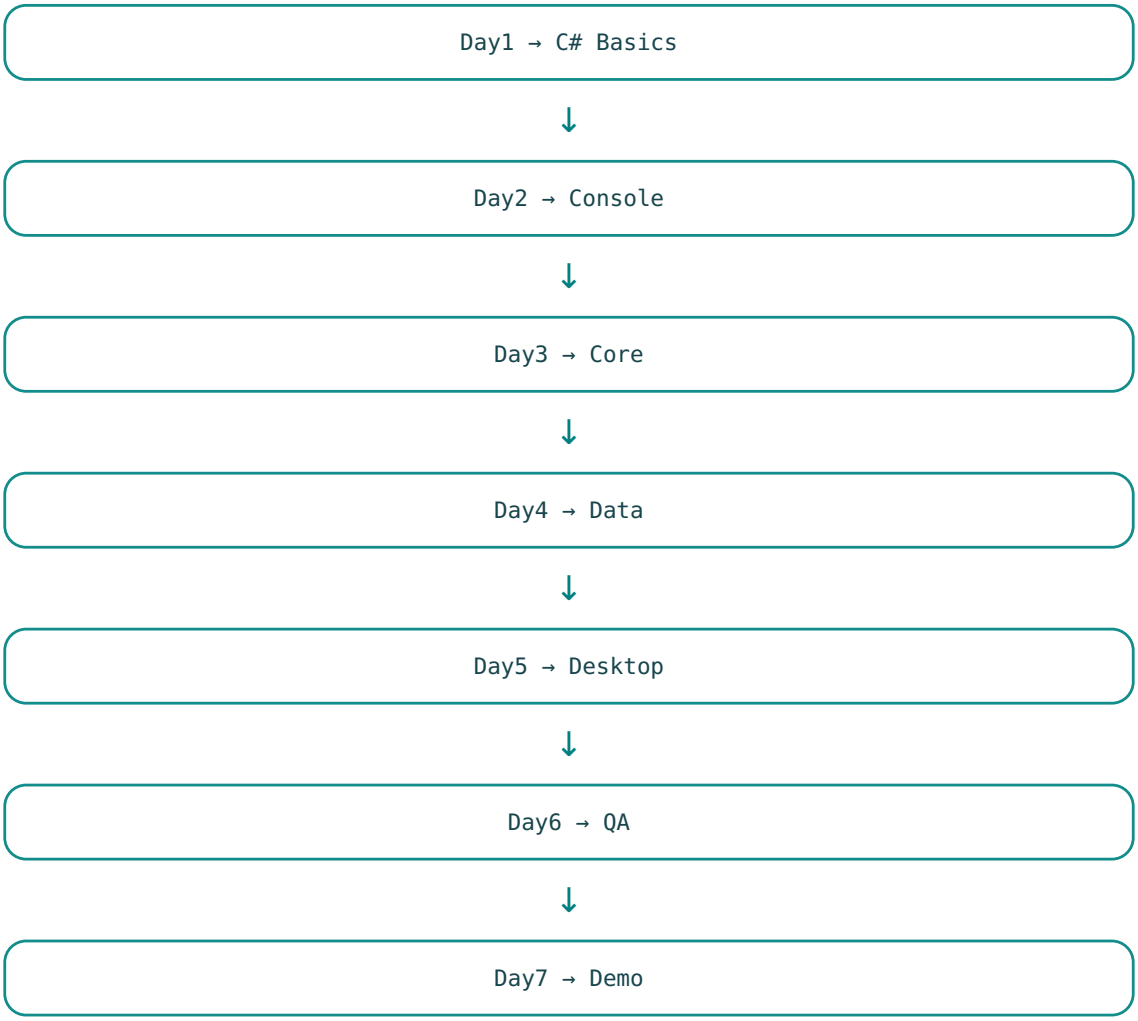
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

docs/BUG_REPORT_TEMPLATE.md.

LESSON 074 / 143

Mini project plan

දින 7ක් තුළ හදන සැලැස්මක්: Day 1 Hello World, Day 2 Console, Day 3 Rules, Day 4 JSON, Day 5 UI, Day 6 Testing, Day 7 Demo.



ඇයි මෙය වැදගත්?

Timebox සහ පොඩි deliverables තිබුණොත් progress පෙනෙනවා.

පුංචි අභ්‍යාසය

ඔබේ පාඩම් දින 7ක් ලෙස calendar එකක සලකුණු කරන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

එක් දවසකට එක් කුඩා milestone එකක්.

LESSON 075 / 143

Expense record නිර්මාණය

Core project එකේ `Expense.cs` file එකට fields පහක් ඇති record එකක් ලියනවා. `Guid` එකෙන් ID එක වෙනස් කර හඳුනාගන්න පුළුවන්.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
public sealed record Expense(
    Guid Id, DateOnly Date,
    string Category, string Description,
    decimal Amount);
```

ඇයි මෙය වැදගත්?

වියදමක් යනු එකම data object එකක් බවට පත් වෙනවා.

පුංචි අභ්‍යාසය

Record එකේ amount type එක `decimal` වීමට හේතුව ලියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Money calculations සඳහා decimal ගැළපෙන නිසා.

LESSON 076 / 143

Category තෝරන්න

Food, Travel, School, Other යන categories පමණක් භාවිතා කරනවා. එවිට spelling variations නිසා reports අවුල් වෙන්නේ නැහැ.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
public static readonly string[] Categories =
    ["Food", "Travel", "School", "Other"];
```

ඇයි මෙය වැදගත්?

Fixed list එකක් තිබීම data quality වැඩි කරනවා.

පුංචි අභ්‍යාසය

"School" සහ "school" එකම value ද? Case-sensitive check ගැන සිතන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

මේ sample validation එකේ නිවැරදි spelling/case අවශ්‍යයි.

LESSON 077 / 143

Manager කියන මැදිහත්කරු

ExpenseManager එකේ records තබාගෙන Add, Delete, Find සහ total ගණනය කරන methods තියෙනවා. එය UI එකෙන් වෙන්වී තියෙනවා.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
private readonly List<Expense> _expenses = [];
public IReadOnlyList<Expense> All =>
    _expenses.AsReadOnly();
```

ඇයි මෙය වැදගත්?

Business logic එක single location එකක තිබීම bug fix පහසු කරනවා.

පුංචි අභ්‍යාසය

Records list එකට outside code එකෙන් අහඹුව වෙනස්කම් කළ නොහැකි වීමේ වාසිය කියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Validation bypass වීම අඩු වෙනවා.

LESSON 078 / 143

Add method flow

Add method එක Date, Category, Description, Amount ලබාගෙන valid නම් අලුත් Expense object එකක් හදලා list එකට එකතු කරනවා.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
Expense item = new(
    Guid.NewGuid(), date, category, description, amount);
_expenses.Add(item);
return item;
```

ඇයි මෙය වැදගත්?

Button click එකේ rules නොලියා Core method එකක් call කිරීම හොඳයි.

පුංචි අභ්‍යාසය

Add method එකෙන් returned value එක කුමක්ද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

අලුත් Expense item එක.

LESSON 079 / 143

Amount validation

Amount එක 0ට වඩා වැඩි විය යුතුයි. Learning app එකේ upper limit එක Rs 1,000,000. Invalid input එකකට `ArgumentException` දාන්න.

Visual Studio Code / C# .cs

EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
if (amount <= 0m || amount > 1_000_000m)
    throw new ArgumentException(
        "Amount must be more than 0 ...");
```

ඇයි මෙය වැදගත්?

Negative amount වැනි දත්ත save වීම වැළැක්වීම fundamental QA rule එකකි.

පුංචි අභ්‍යාසය

-10, 0, 10, 1000001 අතර වලංගු value කුමක්ද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

10.

LESSON 080 / 143

Description validation

Description එක හිස් නොවිය යුතුයි; අකුරු 1 සිට 80 අතර. `Trim()` මඟින් දෙපස හිස්තැන් ඉවත් කරනවා.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
description = description.Trim();
if (description.Length is < 1 or > 80)
    throw new ArgumentException("Invalid description");
```

ඇයි මෙය වැදගත්?

වෙනත් පැත්තෙන් Space key පමණක් ඔබා record save කිරීම වැළැක්වීමට.

පුංචි අභ්‍යාසය

`" "` පමණක් දුන්නොත් valid ද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

නැහැ; Trim කළ පසු length 0.

LESSON 081 / 143

List එක පෙන්වීම

Console menu එකේ 2 තෝරන විට `manager.Find("", "All")` මඟින් සියලු records ලබාගෙන foreach loop එකෙන් පෙන්වනවා.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
foreach (Expense x in manager.Find("", "All"))
    Console.WriteLine($"{x.Description}: {x.Amount:0.00}");
```

ඇයි මෙය වැදගත්?

List එක sort කර display කිරීම user experience පහසු කරනවා.

පුංචි අභ්‍යාසය

List එක හිස් නම් output එකේ records පෙන්වෙනවාද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

නැහැ; foreach 0 වතාවක් run වෙනවා.

LESSON 082 / 143

Search හා filter

Search text එක description තුළ තිබේද කියා පරීක්ෂා කරනවා. Filter එකෙන් Food පමණක් හෝ School පමණක් තෝරා පෙන්වන්න පුළුවන්.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
var result = manager.Find("book", "School");
foreach (var x in result)
    Console.WriteLine(x.Description);
```

ඇයි මෙය වැදගත්?

User කෙනෙකුට records 100ක් තිබුණත් අවශ්‍ය item එක සොයන්න පුළුවන්.

පුංචි අභ්‍යාසය

School category එකේ Notebook item එක book search එකට ගැළපෙනවාද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

ඔව්, "Notebook" තුළ "book" තිබෙනවා.

LESSON 083 / 143

Delete by Id

Record එකක ID එකෙන් නිශ්චිත item එක හඳුනාගෙන Delete කරන්න. App UI එකේ user confirm කළ පසු පමණක් delete ක්‍රියාව සිදුවෙනවා.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
Expense? item = _expenses.FirstOrDefault(x => x.Id == id);
return item is not null && _expenses.Remove(item);
```

ඇයි මෙය වැදගත්?

අහඹු ලෙස record මකාදැමීම වැළැක්වීමට confirmation සහ accurate ID දෙක වැදගත්.

පුංචි අභ්‍යාසය

Description දෙකම Pen නම් එකක් පමණක් මකාගන්නේ කොහොමද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Unique Id එකෙන්.

LESSON 084 / 143

Month total ගණනය

අදාළ year සහ month එකට ගැළපෙන Expenses පමණක් එකතු කරන්න. `Sum` භාවිතයෙන් decimal values එකතු කළ හැකියි.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
return _expenses
    .Where(x => x.Date.Year == year && x.Date.Month == month)
    .Sum(x => x.Amount);
```

ඇයි මෙය වැදගත්?

September expense එක October total එකට ඇතුළත් නොවිය යුතුයි.

පුංචි අභ්‍යාසය

Oct 100 + Oct 50 + Sep 60 නම් October total කීයද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Rs 150.00.

LESSON 085 / 143

GroupBy Category

`GroupBy(x => x.Category)` මඟින් Food, Travel වැනි groups වෙන් කර ඒවායේ totals ගණනය කරන්න පුළුවන්.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
var totals = _expenses.GroupBy(x => x.Category)
    .ToDictionary(g => g.Key, g => g.Sum(x => x.Amount));
```

ඇයි මෙය වැදගත්?

Simple analysis එකක් ලෙස කුමන category එකට වැඩිපුර වියදම් ද යන්න බැලිය හැකියි.

පුංචි අභ්‍යාසය

Food 100 + Food 200 + Travel 50 නම් Food total කීයද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Rs 300.00.

LESSON 086 / 143

ExpenseStore එක

Data saving Core logic එකෙන් වෙන් කර `ExpenseStore` class එක තුළ තබනවා. එවිට වෙනත් storage ක්‍රමයක් පසුව එකතු කිරීම පහසු වෙනවා.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
ExpenseStore store = new();
store.Save(manager.All);
List<Expense> saved = store.Load();
```

ඇයි මෙය වැදගත්?

Single Responsibility Principle එකට සරල උදාහරණයක් ලෙස මේ class එක බලන්න.

පුංචි අභ්‍යාසය

Load method එකේ return type එක මොකක්ද?

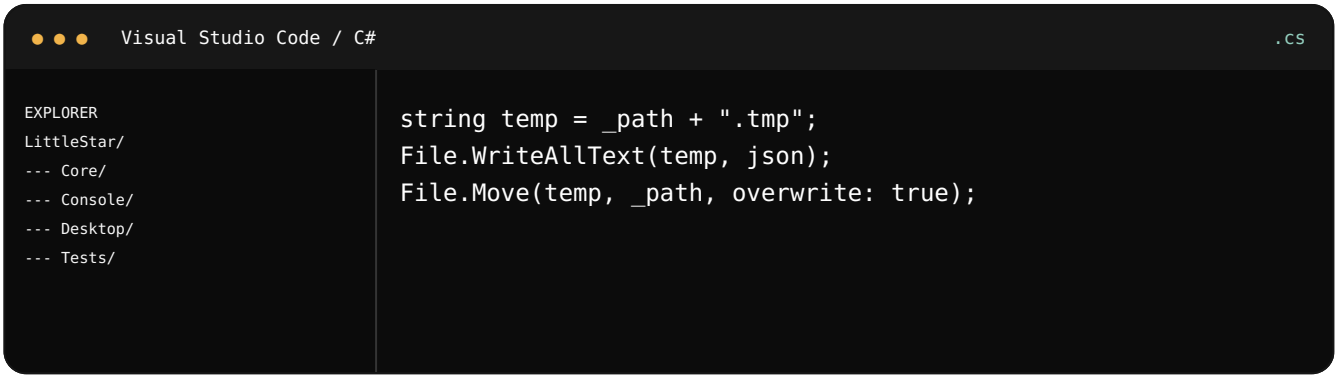
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

List<Expense>.

LESSON 087 / 143

Safe JSON file update

අපේ sample එක අලුත් JSON එක `.tmp` temporary file එකකට ලියා පසුව main file එකට move කරනවා. ඒක direct overwrite එකට වඩා හොඳ ඉගෙනුම් පුරුද්දක්.



ඇයි මෙය වැදගත්?

මේක full database transaction එකක් නොවුණත් අසම්පූර්ණ write එකක අවදානම අඩු කරයි.

පුංචි අභ්‍යාසය

Temporary file එක භාවිතා කරන්නේ ඇයි?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Save process එකේ robustness වැඩි කිරීමට.

LESSON 088 / 143

CSV Export

Data file එක වෙනත් program එකක බලන්න CSV export ලබාදෙනවා. Comma සහ quotes ඇතුළත් description තිබුණත් field එක නිවැරදිව quote කිරීම වැදගත්.

Windows Command Prompt / CMD

```
Date,Category,Description,Amount
2026-10-01,"School","Notebook",250.00
```

ඇයි මෙය වැදගත්?

Data output format එක අනිත් tools සමඟ එකඟවීම interoperability කියන සංකල්පයට උදාහරණයක්.

පුංචි අභ්‍යාසය

CSV file එක Excel වැනි tool එකක open කර columns හරිද බලන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Date, Category, Description, Amount වෙන වෙනම columns.

LESSON 089 / 143

Console project එක Core වෙත සම්බන්ධ කිරීම

Console `.csproj` file එක Core project එකට ProjectReference එකක් දෙනවා. Console project එකට business rules නැවත ලියන්න ඕනේ නැහැ.

Visual Studio Code / C# .cs

EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
<ItemGroup>
 <ProjectReference Include="../LittleStarExpenses.Core/
LittleStarExpenses.Core.csproj" />
</ItemGroup>
```

ඇයි මෙය වැදගත්?

Shared library concept එකේ අරමුණ code duplication අඩු කිරීමයි.

පුංචි අභ්‍යාසය

ProjectReference කියන්නේ කුමක් සඳහාද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

වෙනත් .NET project එකක code භාවිතයට.

LESSON 090 / 143

Console menu එක

Console app එකේ 1 Add, 2 List, 3 Monthly total, 4 Export, 0 Exit කියලා option menu එකක් තියෙනවා. User අංකය type කරනවා.

```
Windows Command Prompt / CMD

1 Add
2 List
3 Monthly total
4 Export CSV
0 Exit
Choose: 1
```

ඇයි මෙය වැදගත්?

පෙළ අඩු ලෙස තබා interface එක පහසු කරගන්න.

පුංචි අභ්‍යාසය

App එකෙන් ඉවත් වීමට තෝරාගන්නා value කුමක්ද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

0.

LESSON 091 / 143

User date input

Date එක `YYYY-MM-DD` ආකාරයෙන් ලබාගන්න. Blank නම් අද දිනය භාවිතා කරනවා. `ParseExact` නිවැරදි format එකක් අවශ්‍ය කරනවා.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
DateOnly date = DateOnly.ParseExact(
    "2026-10-31", "yyyy-MM-dd",
    CultureInfo.InvariantCulture);
```

ඇයි මෙය වැදගත්?

Date format එක user friendly ලෙස prompt එකේම පැහැදිලි කළ යුතුයි.

පුංචි අභ්‍යාසය

2026 December 5 දිනය ඒ format එකෙන් ලියන්න.

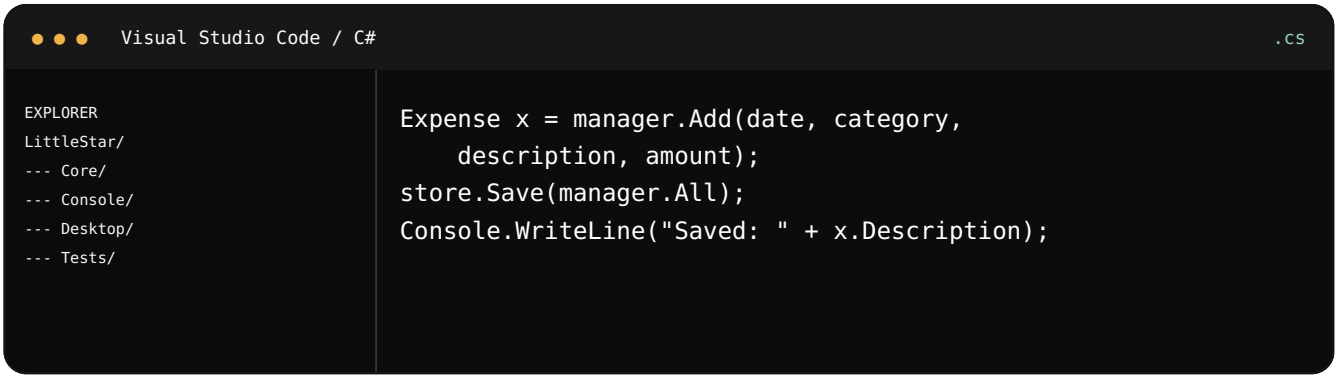
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

2026-12-05.

LESSON 092 / 143

Console Add flow

Console එක amount ලබාගත් පසු `manager.Add(...)` call කර, `store.Save` මඟින් JSON update කරනවා. සාර්ථක නම් Saved message එකක් පෙන්වනවා.



ඇයි මෙය වැදගත්?

Save failure එකක් තිබියදී “Saved” කියලා නොකියන එක වැදගත්.

පුංචි අභ්‍යාසය

store.Save එකට පෙර Console.WriteLine කළොත් ඇතිවන අවදානම කුමක්ද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Save fail වුණත් user success කියා සිතිය හැකියි.

LESSON 093 / 143

Monthly total CMD

Console app එකේ 3 තෝරන විට අද වර්ෂය සහ මාසය manager වෙත යවා total ලබාගන්නවා.

Visual Studio Code / C# .cs

EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
decimal total = manager.TotalForMonth(
    DateTime.Today.Year, DateTime.Today.Month);
Console.WriteLine($"Rs {total:0.00}");
```

ඇයි මෙය වැදගත්?

Month/year argument නිසා tests වල වෙනත් මාස ගැන පරීක්ෂා කරන්න පුළුවන්.

පුංචි අභ්‍යාසය

අද නොවන මාසයක total test කරන්න code එක වෙනස් කරන්නේ කොහොමද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Method එකට වර්ෂය/මාසය සෘජුව යවන්න.

LESSON 094 / 143

CMD error message හොඳ කරමු

Invalid amount, invalid date, file permission error සඳහා friendly message එකක් ලබාදෙන්න. Userට ගැටලුව හදාගන්න පුළුවන් විය යුතුයි.

Windows Command Prompt / CMD

```
Please fix: Amount must be more than 0.
Please fix: Enter a number such as 250.50.
```

ඇයි මෙය වැදගත්?

“Unknown exception” වැනි message වලට වඩා specific help එකක් වැදගත්.

පුංචි අභ්‍යාසය

Invalid data prompt එකක් සඳහා තවත් පැහැදිලි message එකක් ලියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

"Please enter a date like 2026-10-31".

LESSON 095 / 143

Console app run කරන්න

ZIP එක extract කළ පසු top folder එකෙන් `dotnet run --project LittleStarExpenses.Console` run කරන්න. Core project reference එක නිසා code build වෙයි.

Windows Command Prompt / CMD

```
dotnet run --project LittleStarExpenses.Console
=== Little Star Expense Tracker ===
1 Add  2 List  3 Monthly total  4 Export CSV  0 Exit
```

ඇයි මෙය වැදගත්?

Windows Forms app එකට පෙර console flow එකෙන් business logic තේරුම් ගන්න.

පුංචි අභ්‍යාසය

1 Add, 2 List, 3 Monthly total ක්‍රියාවලිය පිළිවෙළින් කරන්න.

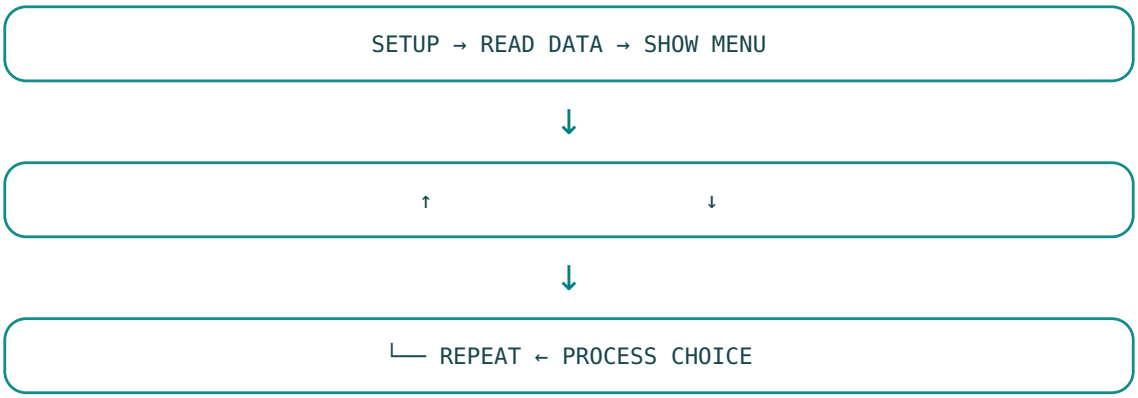
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

අලුත් record එක list සහ monthly total තුළ පෙන්විය යුතුයි.

LESSON 096 / 143

Console code කියවීමේ ක්‍රමය

`Program.cs` හි top-level statements, while loop, user input, try/catch සහ save එක එකට සම්බන්ධ වෙනවා. එකවර මුළු file එක memorize කරන්න එපා.



ඇයි මෙය වැදගත්?

Code එකේ sections වෙන් කර බලන එක නවකයන්ට ප්‍රයෝජනවත්.

පුංචි අභ්‍යාසය

While loop එකේ exit condition එක පෙන්වන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

choice == "0" නම් break.

LESSON 097 / 143

WinForms project එක

`dotnet new winforms -n LittleStarExpenses.Desktop -f net10.0` command එක Windows මත භාවිතා කළ හැකියි. අපේ ZIP එකේ එය සකස් කර තිබෙනවා.

Windows Command Prompt / CMD

```
dotnet run --project LittleStarExpenses.Desktop
```

ඇයි මෙය වැදගත්?

Desktop app එක build/run කිරීමට Windows අවශ්‍යයි; Linux/macOS සඳහා WinForms නොවේ.

පුංචි අභ්‍යාසය

මේ app එක Linux පරිගණකයක direct run නොකළ හැක්කේ ඇයි?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Windows Forms Windows-specific UI framework එකක් නිසා.

LESSON 098 / 143

Program.cs startup

Windows Forms app එකේ `[STAThread]` සහ `Application.Run(new MainForm())` මඟින් window එක පටන්ගන්නවා.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
[STAThread]
static void Main()
{
  ApplicationConfiguration.Initialize();
  Application.Run(new MainForm());
}
```

ඇයි මෙය වැදගත්?

Console.WriteLine වෙනුවට UI window එක event loop තුළ run වෙනවා.

පුංචි අභ්‍යාසය

MainForm class එකේ අරමුණ මොකක්ද?

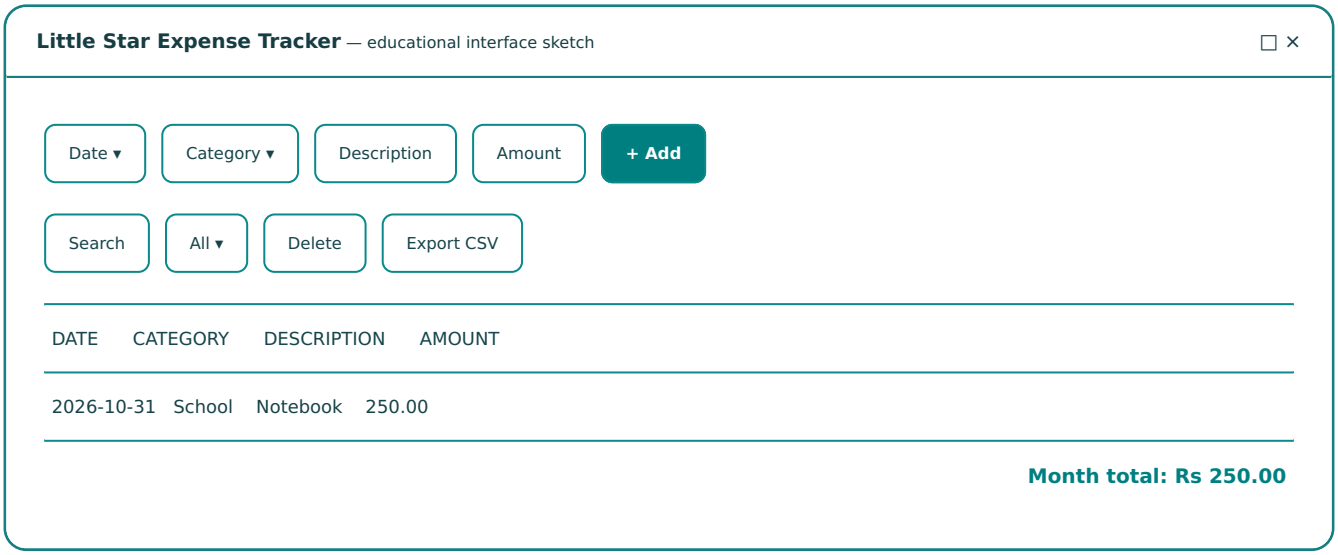
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

App එකේ ප්‍රධාන window එක.

LESSON 099 / 143

Form එක යනු window එක

MainForm එක Windows screen එකේ open වෙන application window එකයි. Title, Size, Font වැනි properties වලින් පෙනුම වෙනස් කරනවා.



ඇයි මෙය වැදගත්?

අකුරු කියවීමට පහසු ලෙස ලොකු font සහ white background තෝරාගත්තා.

පුංචි අභ්‍යාසය

Window එකේ title එක වෙනස් කරන property එක කුමක්ද?

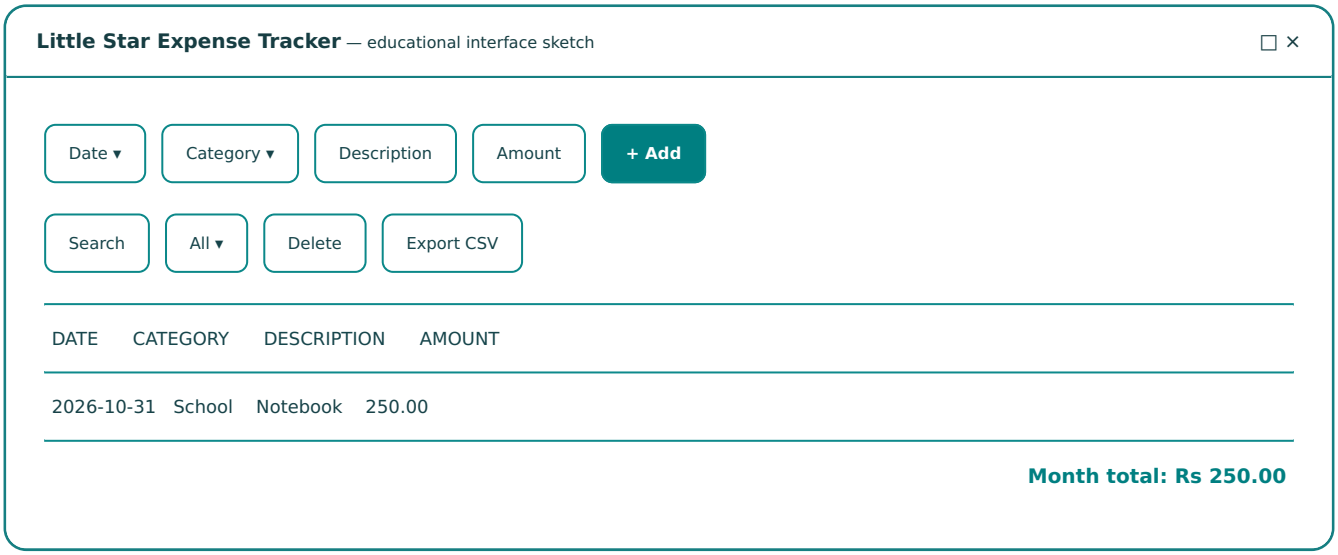
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Text.

LESSON 100 / 143

Label සහ TextBox

Label එක දත්ත ඇතුළත් කිරීමට කියන නම පෙන්වනවා. TextBox එකෙන් description හෝ amount ලබාගන්නවා.



ඇයි මෙය වැදගත්?

User‍ට input fields වල අර්ථය පැහැදිලිව දැක්විය යුතුයි.

පුංචි අභ්‍යාසය

PlaceholderText සහ actual value එකේ වෙනස කියන්න.

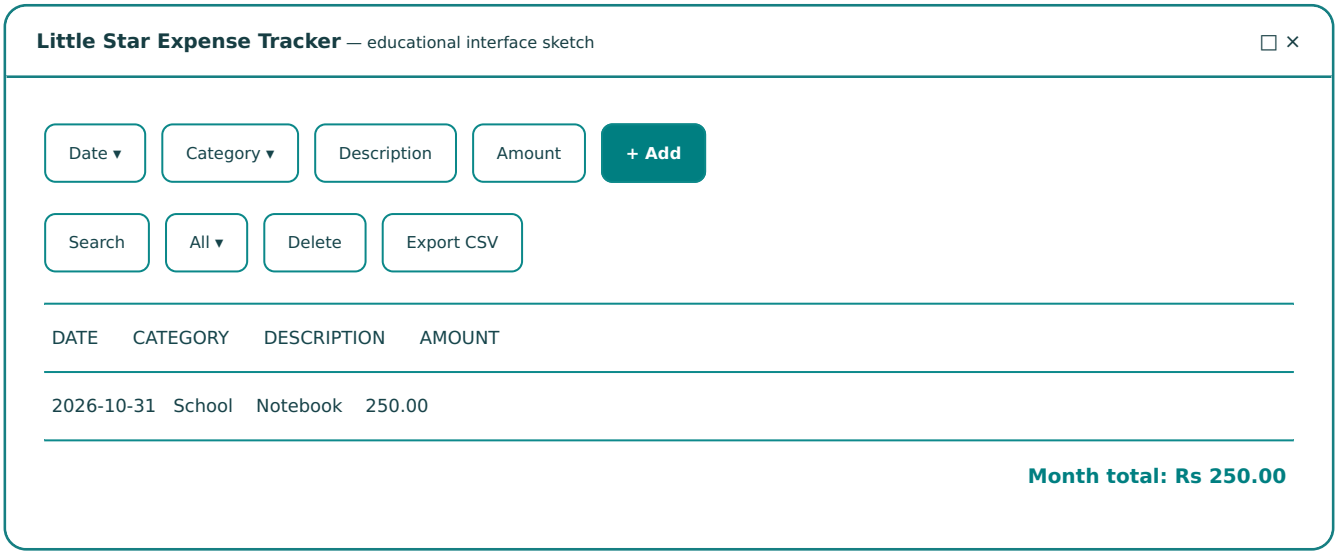
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Placeholder එක hint එකක්; actual value user type කරනවා.

LESSON 101 / 143

ComboBox category

ComboBox එකේ Food, Travel, School, Other යන options තෝරාගන්න පුළුවන්. Userට අහඹු spelling එකක් type කරන්න ඉඩ නොදෙනවා.



ඇයි මෙය වැදගත්?

Fixed list එකක් data consistency තබයි.

පුංචි අභ්‍යාසය

DropDownList style එකෙන් ලැබෙන වාසිය කුමක්ද?

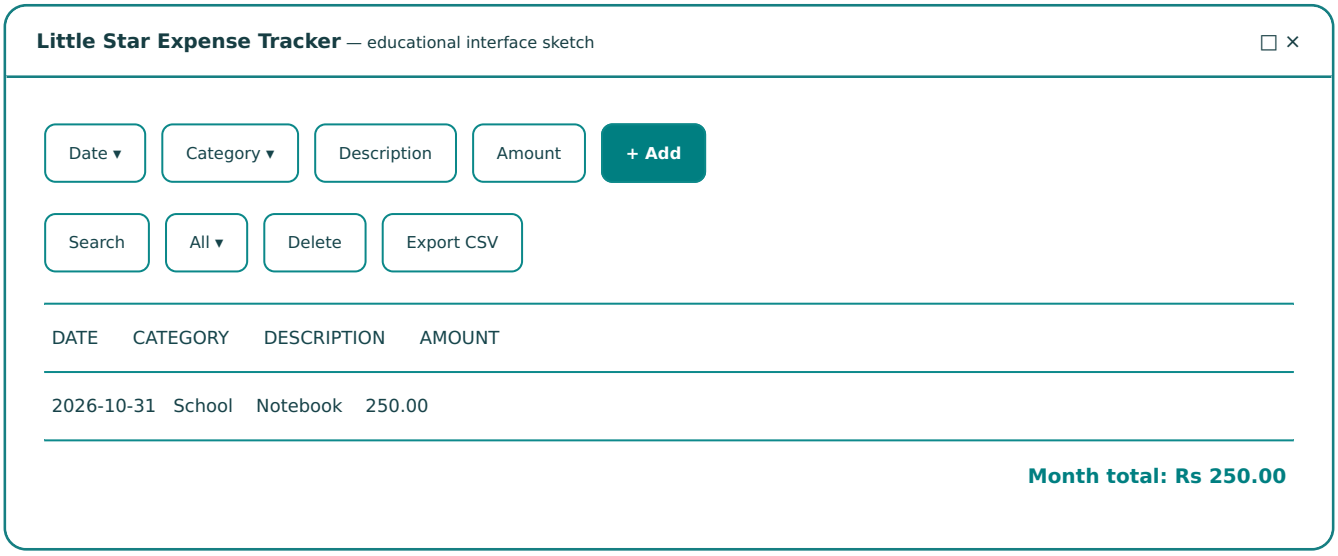
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

User අනුමත නොකළ category text type නොකරයි.

LESSON 102 / 143

DateTimePicker

Userට calendar එකෙන් date තෝරන්න DateTimePicker දෙනවා. එහි selected value එක DateOnly බවට පරිවර්තනය කරනවා.



ඇයි මෙය වැදගත්?

Date වැරදි format එකක type කිරීමේ අවදානම අඩුවෙනවා.

පුංචි අභ්‍යාසය

DateTimePicker එක තෝරාගන්නා දේ කුමක්ද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

දිනය (calendar date).

LESSON 103 / 143

Button සහ Click event

Add button එක click කළ විට `AddExpense()` method එක call කරනවා. මෙය event-driven programming කියන සංකල්පයට උදාහරණයක්.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
Button add = new() { Text = "+ Add expense" };
add.Click += (_, _) => AddExpense();
```

ඇයි මෙය වැදගත්?

User action එකක් මත code run වෙන බව පැහැදිලි වෙනවා.

පුංචි අභ්‍යාසය

Click event එකක් නොමැති button එකක් ඔබන විට මොකද වෙන්නේ?

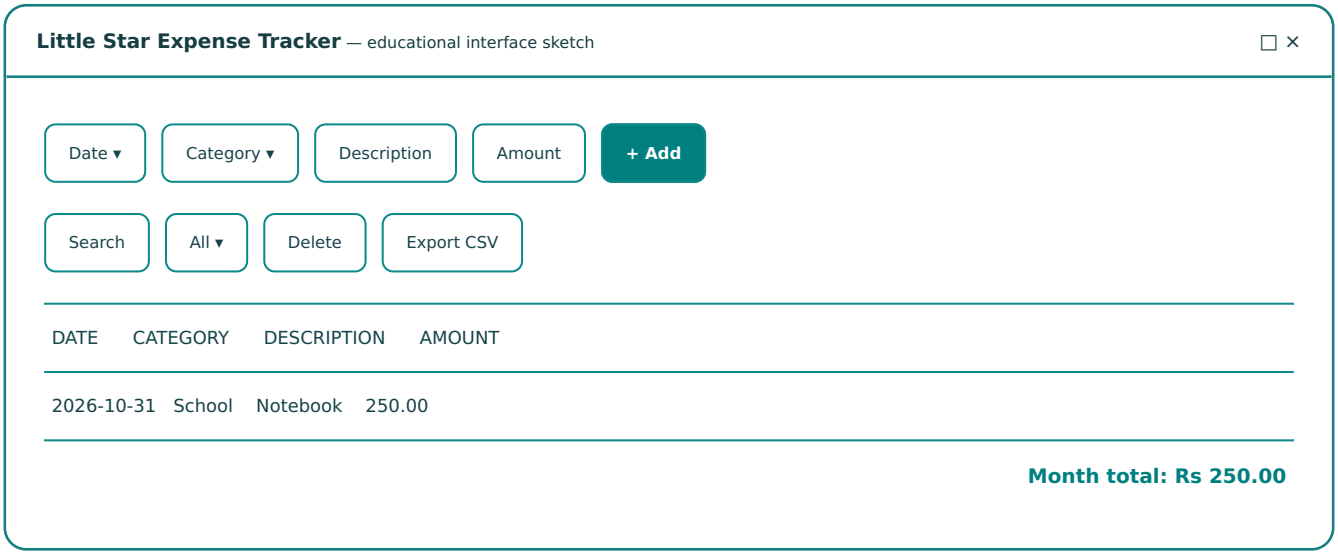
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

අදාළ action එකක් නොපැවරූ විට කිසිවක් නොවිය හැකියි.

LESSON 104 / 143

DataGridView table

Expense records පේළි සහ columns ලෙස පෙන්වීමට DataGridView භාවිතා කරනවා. Date, Category, Description, Amount යන columns සකස් කරනවා.



ඇයි මෙය වැදගත්?

Table එක read-only කරලා edits manager rules හරහා පමණක් කිරීමට දීලා තියෙනවා.

පුංචි අභ්‍යාසය

ReadOnly = true වීමේ වාසිය මොකක්ද?

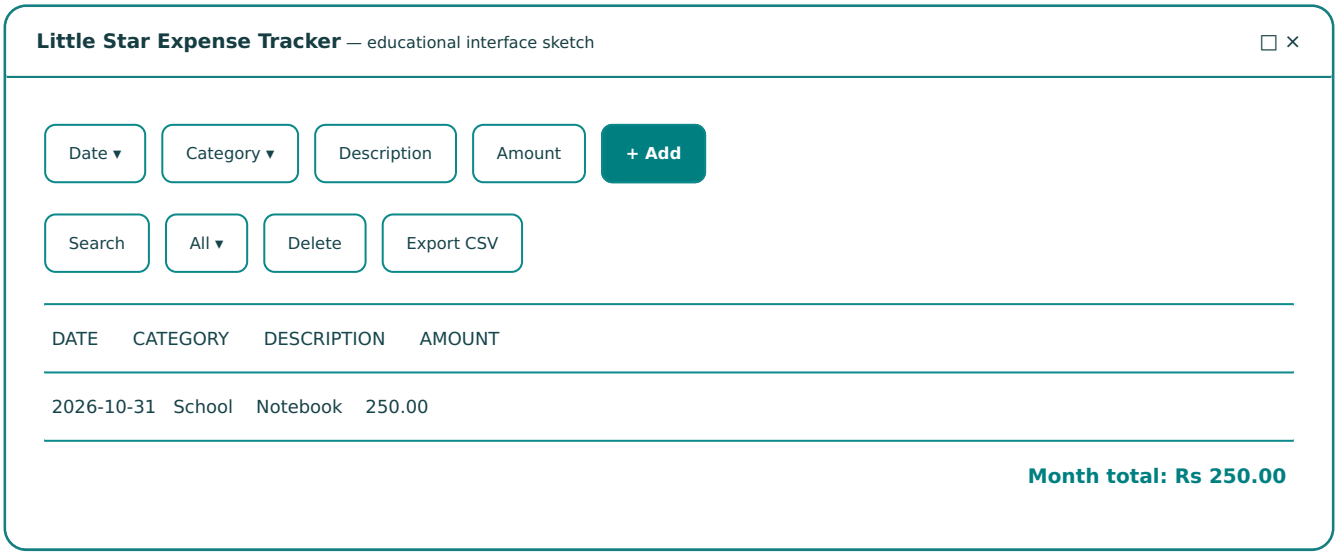
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Table තුළින් validation bypass කර edit කිරීම අඩු වෙනවා.

LESSON 105 / 143

Window layout සැලසුම

Input row එක, action/filter row එක, data table එක යන කොටස් තුන window එකේ තියෙනවා. Dock සහ FlowLayoutPanel මඟින් controls පෙළගස්වනවා.



ඇයි මෙය වැදගත්?

Window resize කළත් table එක ඉතිරි ඉඩ පිරවීම user friendly.

පුංචි අභ්‍යාසය

Form එක තුළ table එකට වැඩිපුර ඉඩ දෙන්නේ ඇයි?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Records ගොඩක් එකවර බලන්න.

LESSON 106 / 143

Add button validation

UI එකේ amount text එක decimal බවට parse කරන්න. Parse අසාර්ථක නම් friendly message එකක් පෙන්වා return කරන්න.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
if (!decimal.TryParse(amount.Text, out decimal value))
{
  MessageBox.Show("Enter a valid amount");
  return;
}
```

ඇයි මෙය වැදගත්?

User error එකක් exception එකකට පත්කරන්නට පෙර check කිරීම හොඳයි.

පුංචි අභ්‍යාසය

"abc" type කර Add කළොත් කුමක් විය යුතුද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Error message එකක්; record save නොවේ.

LESSON 107 / 143

Save + Refresh

AddExpense method එක manager.Add, store.Save සහ RefreshTable call කරනවා. එවිට අලුත් expense එක table එකේ පෙන්වෙයි.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
manager.Add(dateValue, category.Text,
    description.Text, value);
store.Save(manager.All);
RefreshTable();
```

ඇයි මෙය වැදගත්?

Data save කරලා success state පෙන්වීම user feedback සඳහා වැදගත්.

පුංචි අභ්‍යාසය

Save අසාර්ථක නම් පෙන්විය යුතු message එක කුමක්ද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

"Save failed" වැනි friendly message එකක්.

LESSON 108 / 143

Search box එක

User description එකේ කොටසක් type කරන විට TextChanged event එකෙන් table එක අලුත් වෙනවා.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
search.TextChanged += (_, _) => RefreshTable();
```

ඇයි මෙය වැදගත්?

Search results ඉක්මනින් දකින්න පුළුවන් නිසා usability වැඩි වෙනවා.

පුංචි අභ්‍යාසය

Notebook දත්තයක් සෙවීමට "note" type කළොත් පෙන්වෙනවාද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

ඔව්, case-insensitive substring search.

LESSON 109 / 143

Category Filter

Dropdown එකෙන් All/Food/School ආදී options තෝරා table records filter කරන්න. Filter එක වෙනස් වූ වහා table එක refresh කරන්න.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
filter.SelectedIndexChanged += (_, _) => RefreshTable();
var rows = manager.Find(search.Text, filter.Text);
```

ඇයි මෙය වැදගත්?

Search සහ category එකට භාවිතා කළ හැකි බව practical app feature එකක්.

පුංචි අභ්‍යාසය

All තෝරාගත් විට කුමන category records පෙන්වයිද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

හැම category එකම.

LESSON 110 / 143

Delete confirmation

Delete Selected click කළ විට selected Id එක හඳුනාගෙන "Delete this sample record?" කියලා අසනවා. Yes නම් Delete කර Save කරනවා.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
if (MessageBox.Show("Delete?", "Confirm",
    MessageBoxButtons.YesNo) != DialogResult.Yes)
    return;
```

ඇයි මෙය වැදගත්?

Destructive action එකකට confirmation dialog එක වැදගත්.

පුංචි අභ්‍යාසය

No තෝරාගත් විට record එකට කුමක් විය යුතුද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Record එක එලෙසම තිබිය යුතුයි.

LESSON 111 / 143

CSV Export dialog

SaveFileDialog එකෙන් user කැමති folder සහ filename එක තෝරා ගන්නවා. ඉන්පසු CSV content එක file එකක් ලෙස save කරනවා.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
using SaveFileDialog dialog = new() {
    FileName = "little-star-expenses.csv",
    Filter = "CSV file (*.csv)|*.csv"
};
```

ඇයි මෙය වැදගත්?

Userගේ files override නොකර සුරකීමට dialog එක උපකාරී.

පුංචි අභ්‍යාසය

Export වූ file එක Excel තුළ columns 4කට පෙන්වෙනවාද පරීක්ෂා කරන්න.

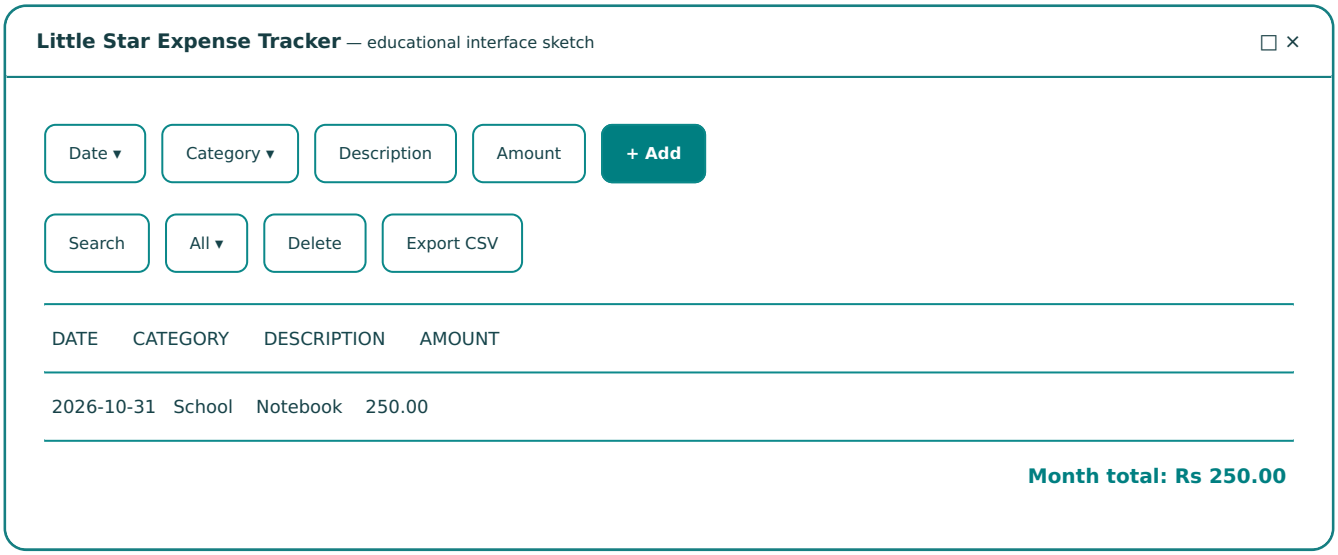
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Date, Category, Description, Amount.

LESSON 112 / 143

Desktop preview එක

මෙම පිටුවේ තියෙන්නේ sample UI sketch එකක්. සැබෑ app එක Windows හි run කළ විට installed fonts සහ window theme අනුව වෙනස් පෙනුමක් තිබිය හැකියි.



ඇයි මෙය වැදගත්?

Sketch එක සහ real screenshot එක වෙනස් බව පැහැදිලි කර දීම නිවැරදි documentation පුරුද්දකි.

පුංචි අභ්‍යාසය

UI sketch එකේ userට කරන්න පුළුවන් ක්‍රියා 4ක් ලියන්න.

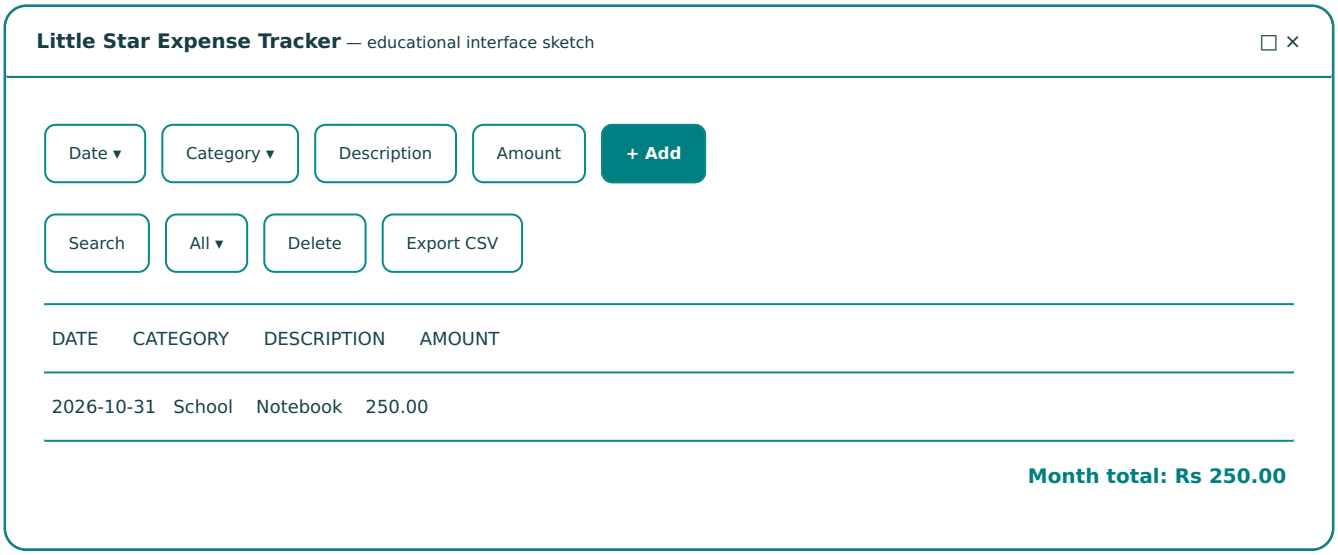
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Add, Search, Delete, Export.

LESSON 113 / 143

Keyboard හා Accessibility

App එක mouse එකට පමණක් සීමා නොවිය යුතුයි. Tab key එකෙන් controls අතර යාම, readable labels, error dialog පෙන්වීම වැදගත්.



ඇයි මෙය වැදගත්?

Accessibility කියන්නේ වැඩි පිරිසකට software භාවිතා කළ හැකි ලෙස නිර්මාණය කිරීමයි.

පුංචි අභ්‍යාසය

Userට mouse නැතිව Amount field එකට යා හැකිද පරීක්ෂා කරන්න.

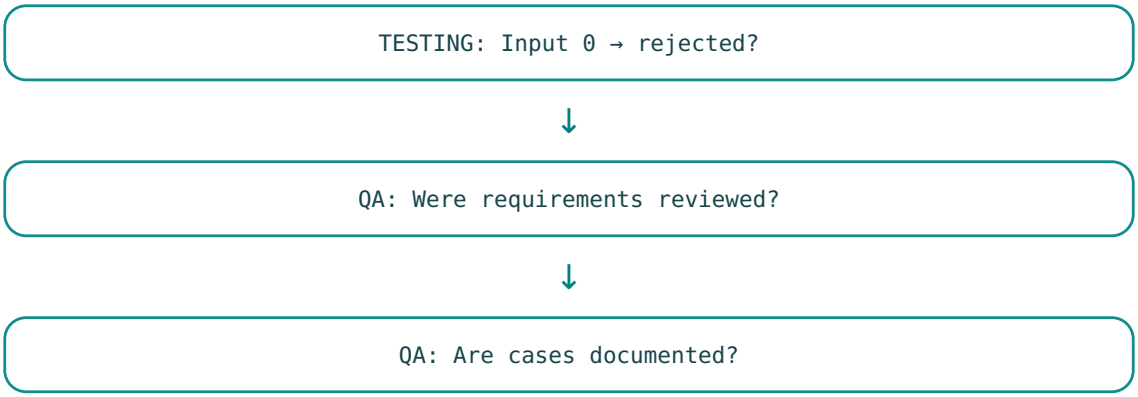
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Tab navigation පරීක්ෂා කරන්න.

LESSON 114 / 143

Testing සහ QA අතර වෙනස

Testing කියන්නේ code එකේ වැඩ නිවැරදිද කියා පරීක්ෂා කිරීම. QA කියන්නේ මුළු development process එකේ quality තහවුරු කිරීම.



ඇයි මෙය වැදගත්?

Code run වූ පමණින් app එක සම්පූර්ණ හොඳයි කියලා නොසිතන්න.

පුංචි අභ්‍යාසය

"Bug report සකස් කිරීම" Testing ද QA process එකේ කොටසක්ද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Quality assurance process එකේ කොටසක්.

LESSON 115 / 143

Test case ලියමු

Test case එකකට ID, Steps, Input, Expected Result, Actual Result සහ Status තියෙනවා. Expected එක code run කිරීමට පෙර ලියන්න.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
TC-02
Input: amount = 0
Expected: Error, no new record
Actual: [fill after running]
```

ඇයි මෙය වැදගත්?

Expected සහ Actual වෙනස් නම් bug එකක් තිබිය හැකියි.

පුංචි අභ්‍යාසය

TC-03 සඳහා negative input test case එකක් ලියන්න.

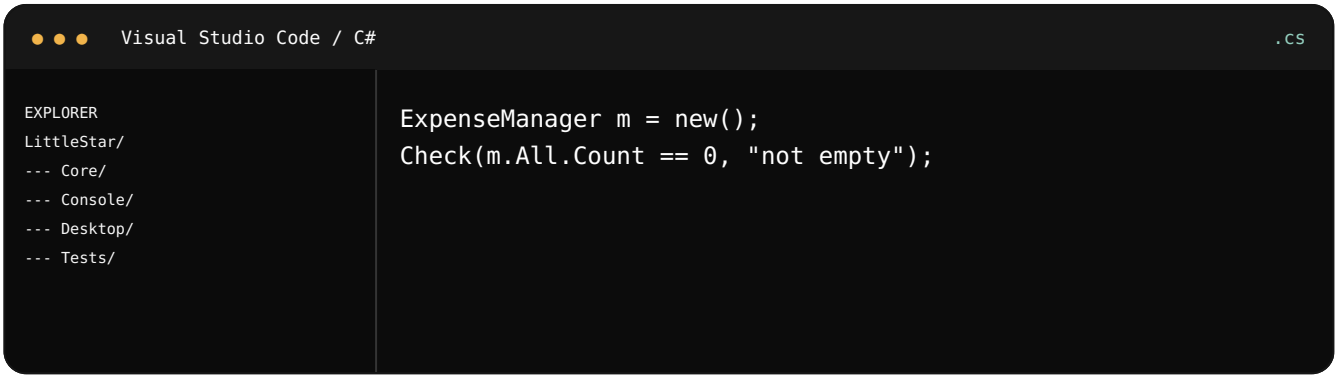
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

-50 → reject; no record saved.

LESSON 116 / 143

Unit test

Unit test යනු තනි method/class logic එක පරීක්ෂා කරන කුඩා program එකක්. අපේ project එකේ external test packages නැති simple test runner එකක් තියෙනවා.



ඇයි මෙය වැදගත්?

Code වෙනස් කරන විට වැරදි අලුත් වෙලාද කියා ඉක්මනින් බලන්න පුළුවන්.

පුංචි අභ්‍යාසය

නව ExpenseManager එකේ මුල් count එක කීයද?

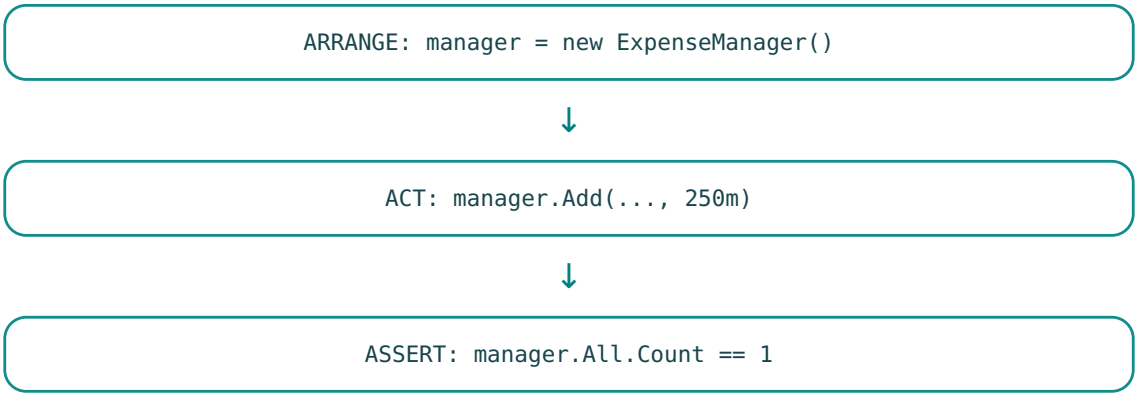
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

0.

LESSON 117 / 143

AAA pattern

Arrange → Act → Assert යනු test case එකක් ලියන හොඳ ආකාරයකි: Data සකස් කරන්න, action එක කරන්න, expected result එක පරීක්ෂා කරන්න.



ඇයි මෙය වැදගත්?

Tests කියවන්න පහසු වන නිසා bugs හඳුනාගැනීමද පහසුයි.

පුංචි අභ්‍යාසය

AAA කියන අකුරු තුනේ තේරුම ලියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Arrange, Act, Assert.

LESSON 118 / 143

Boundary testing

Amount 0, 0.01, 1,000,000, 1,000,000.01 වගේ edge cases පරීක්ෂා කරන්න.

Windows Command Prompt / CMD

```
0.00      → Reject
0.01      → Accept
1,000,000.00 → Accept
1,000,000.01 → Reject
```

ඇයි මෙය වැදගත්?

වැරදි ගොඩක් limits අසල හමුවෙන නිසා boundary tests වැදගත්.

පුංචි අභ්‍යාසය

මෙම values 4න් valid දෙක මොනවාද?

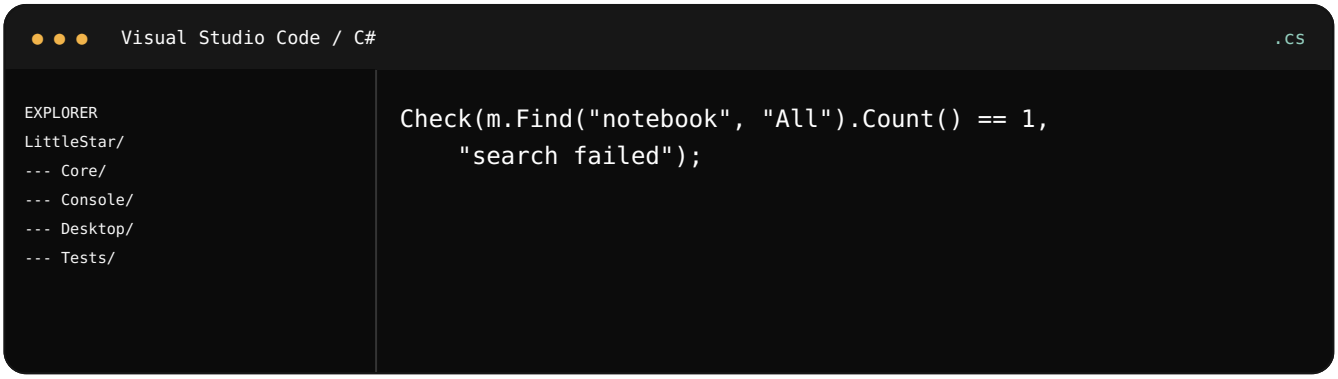
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

0.01 සහ 1,000,000.00.

LESSON 119 / 143

Search unit test

"NOTEBOOK" කියන description එක "notebook" කියලා search කළ විට පෙන්විය යුතුයි. Case-insensitive matching පරීක්ෂා කරන්න.



ඇයි මෙය වැදගත්?

Search logic එක UI එකක් නොමැතිව Core test එකකින් පරීක්ෂා කළ හැකියි.

පුංචි අභ්‍යාසය

"note" කියන කොටස දුන්නත් matching විය යුතුද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

ඔව්, substring match.

LESSON 120 / 143

Monthly total unit test

October 100 සහ September 60 තිබේ නම් October total Rs 100 විය යුතුයි. Date filter එක නිවැරදිද බලන්න.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
Check(m.TotalForMonth(2026, 10) == 100m,
    "wrong total");
```

ඇයි මෙය වැදගත්?

Date boundaries පරීක්ෂා නොකළොත් report එක වැරදි වෙන්න පුළුවන්.

පුංචි අභ්‍යාසය

November 1 record එක October total එකට ඇතුළත් වෙයිද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

නැහැ.

LESSON 121 / 143

JSON round-trip test

Expense එකක් JSON file එකට Save කර නැවත Load කිරීම round-trip test එකක්. එහි data අඩු වී හෝ වෙනස් වී නැතිද බලන්න.

Visual Studio Code / C# .cs

```
EXPLORER
LittleStar/
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
store.Save(manager.All);
var loaded = store.Load();
Check(loaded.Count == 1, "load failed");
```

ඇයි මෙය වැදගත්?

Test එක තාවකාලික folder/file එකක් භාවිතා කළ යුතුයි; real user data වෙත ලිවීම වැරදියි.

පුංචි අභ්‍යාසය

Save සහ Load දෙකම test කිරීමෙන් ලැබෙන වාසිය මොකක්ද?

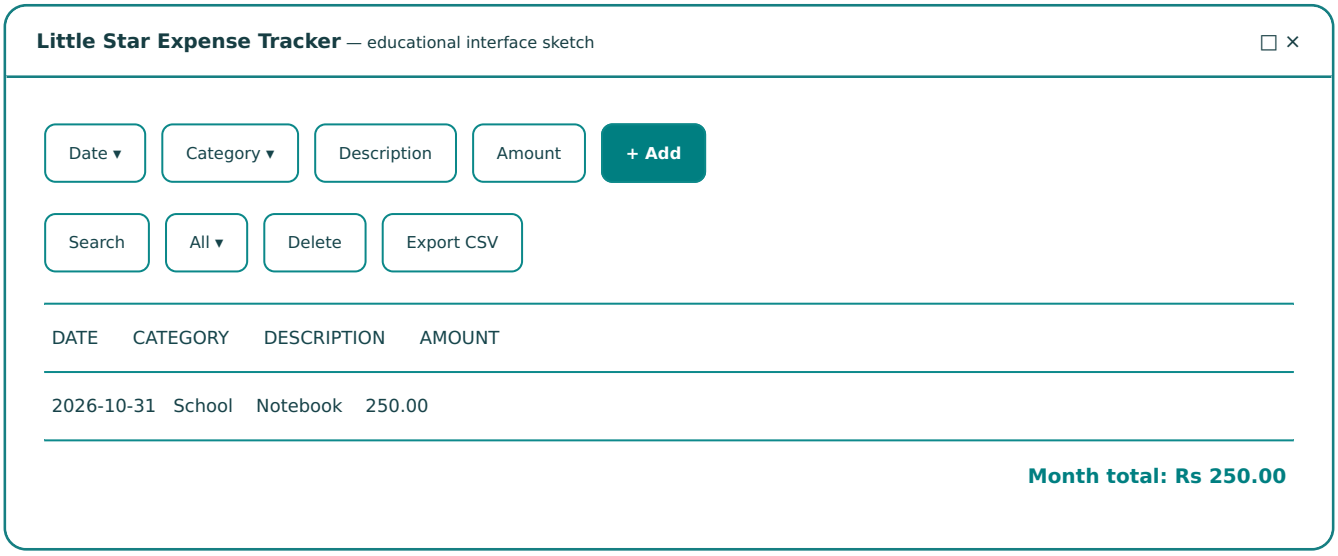
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Persistence එක නිවැරදිද තහවුරු වෙනවා.

LESSON 122 / 143

Manual UI testing

Windows desktop app එකක් භාවිතා කරන ආකාරයට tester කෙනෙකු click, type, filter, export කර බලන්න ඕනේ. Automation tests ඒ සියල්ල වෙනුවට නොපවතිනවා.



ඇයි මෙය වැදගත්?

User කෙනෙක් හිතන විදිහට flow එක check කිරීම අත්‍යවශ්‍යයි.

පුංචි අභ්‍යාසය

App එක reopen කළ පසු record එක තිබිය යුතුද?

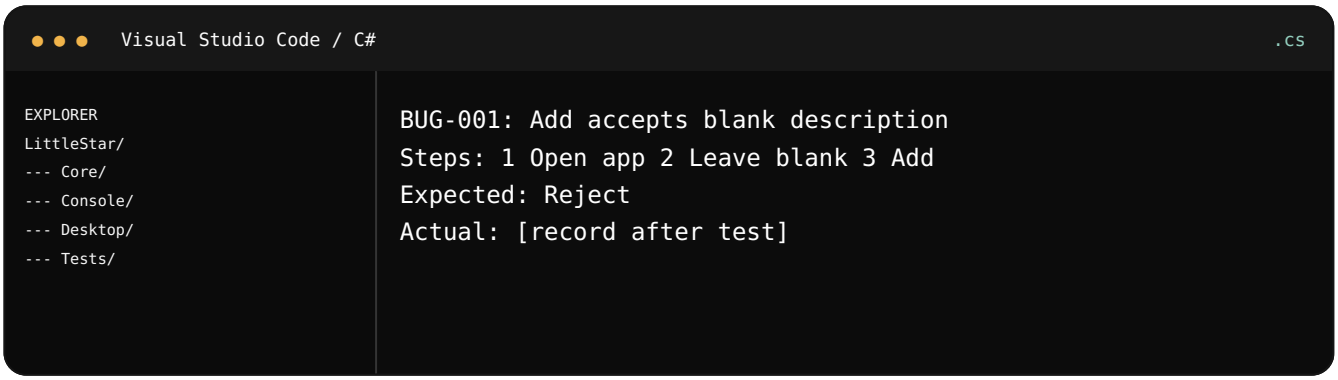
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

ඔව්, JSON saving සාර්ථක නම්.

LESSON 123 / 143

Bug report එක

Bug report එකට Title, Steps, Expected, Actual, Severity, Environment සහ Status දාන්න. Developerට එය නැවත replicate කරගැනීමට හැකි විය යුතුයි.



ඇයි මෙය වැදගත්?

"App not working" කියන report එකට වඩා exact steps හොඳයි.

පුංචි අභ්‍යාසය

මෙහි severity එක තීරණය කිරීමට වැදගත් කරුණක් කියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

User data quality/impact.

LESSON 124 / 143

Debugging පළමු පියවර

Bug එකක් ඇතිවිට reproduce, isolate, inspect, fix, retest කියන පියවර අනුගමනය කරන්න. Error message කියවා line එකට යන්න.

```
REPRODUCE → ISOLATE → FIX → RETEST
```

ඇයි මෙය වැදගත්?

Random code මකාදැමීමෙන් වඩා මේ ක්‍රමය වේගවත්.

පුංචි අභ්‍යාසය

Fix කළ පසුව නැවත test කිරීම අත්‍යවශ්‍ය වන්නේ ඇයි?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Bug එක ඇත්තටම විසඳී ඇතිද බලන්න.

LESSON 125 / 143

Regression testing

නව Search feature එකක් එකතු කළ පසු Add, Save, Delete වගේ පරණ features තවම වැඩ කරනවාද බලන්න. ඒක regression testing.

Windows Command Prompt / CMD

```
After search change:
[ ] Add works
[ ] Save/reopen works
[ ] Delete works
[ ] Total correct
```

ඇයි මෙය වැදගත්?

New feature එකක් පැරණි function එකක් බිඳ දැමිය හැකියි.

පුංචි අභ්‍යාසය

Regression checklist එකට තවත් feature එකක් එකතු කරන්න.

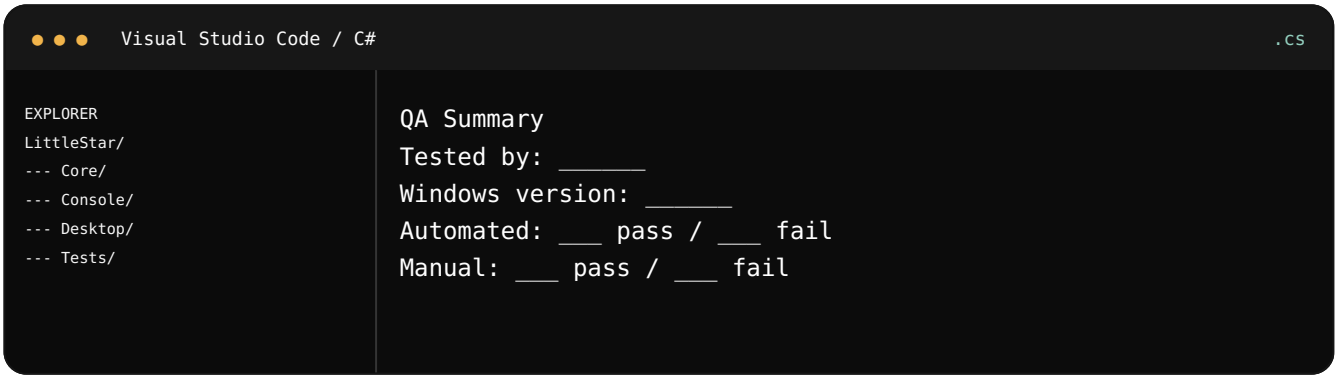
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Export CSV works.

LESSON 126 / 143

QA report හා sign-off

සත්‍ය පරීක්ෂණය කළ පසු පමණක් pass/fail සටහන් කරන්න. QA report එකේ known risks සහ open defects ද සඳහන් කරන්න.



ඇයි මෙය වැදගත්?

අපේ ZIP එකේ templates තිබුණත් ඔබ Windows මත tests run කර actual results පිරවිය යුතුයි.

පුංචි අභ්‍යාසය

Test නොකර "PASS" කියා ලියන එක හරිද?

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

නැහැ.

LESSON 127 / 143

Release checklist

App එක userට දීමට පෙර build, tests, README, data privacy note, backups සහ demo run සම්පූර්ණද බලන්න.

Windows Command Prompt / CMD

```
[ ] dotnet build
[ ] run console and desktop
[ ] automated + manual tests
[ ] README and limitations
[ ] demo data only
```

ඇයි මෙය වැදගත්?

Release කියන්නේ source file copy කිරීම පමණක් නොව reliable handover එකකි.

පුංචි අභ්‍යාසය

Release checklist එකට එකතු කළ යුතු වැදගත් check එකක් ලියන්න.

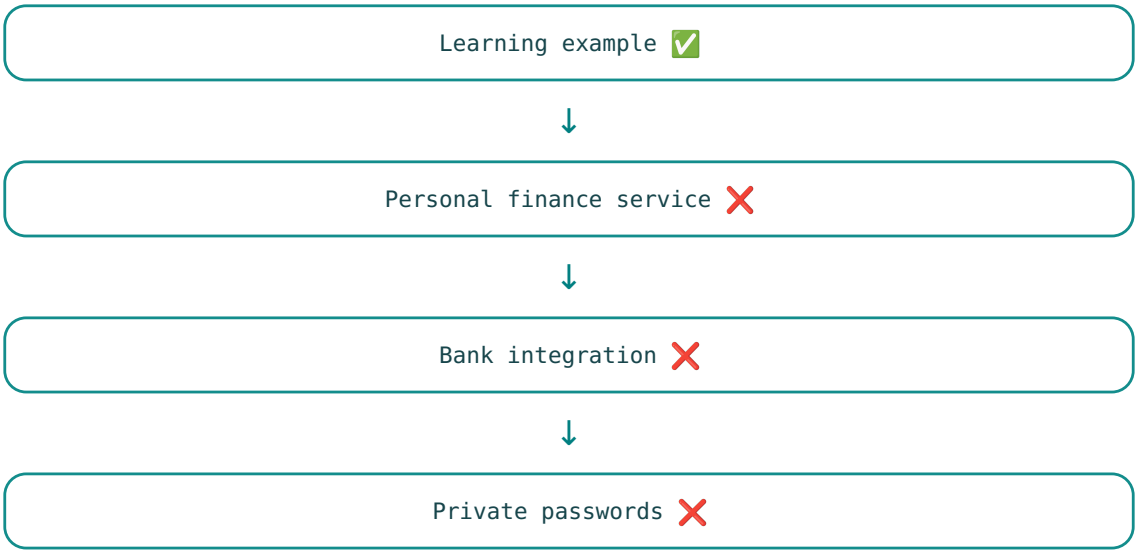
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

උදාහරණ: error messages verified.

LESSON 128 / 143

Responsible coding සහ real-world scope

මෙය real bank app එකක් නොවේ. Login, encryption, compliance, multiuser storage සහ server security මෙහි නැහැ. දරුවන් සඳහා safe test data පමණක් භාවිතා කරන්න.



ඇයි මෙය වැදගත්?

Education project සීමා සහ අරමුණු පැහැදිලිව කියන්නේ හොඳ developer කෙනෙක්.

පුංචි අභ්‍යාසය

මෙම app එකට සැබෑ bank passwords දීම සුදුසු නැත්තේ ඇයි?

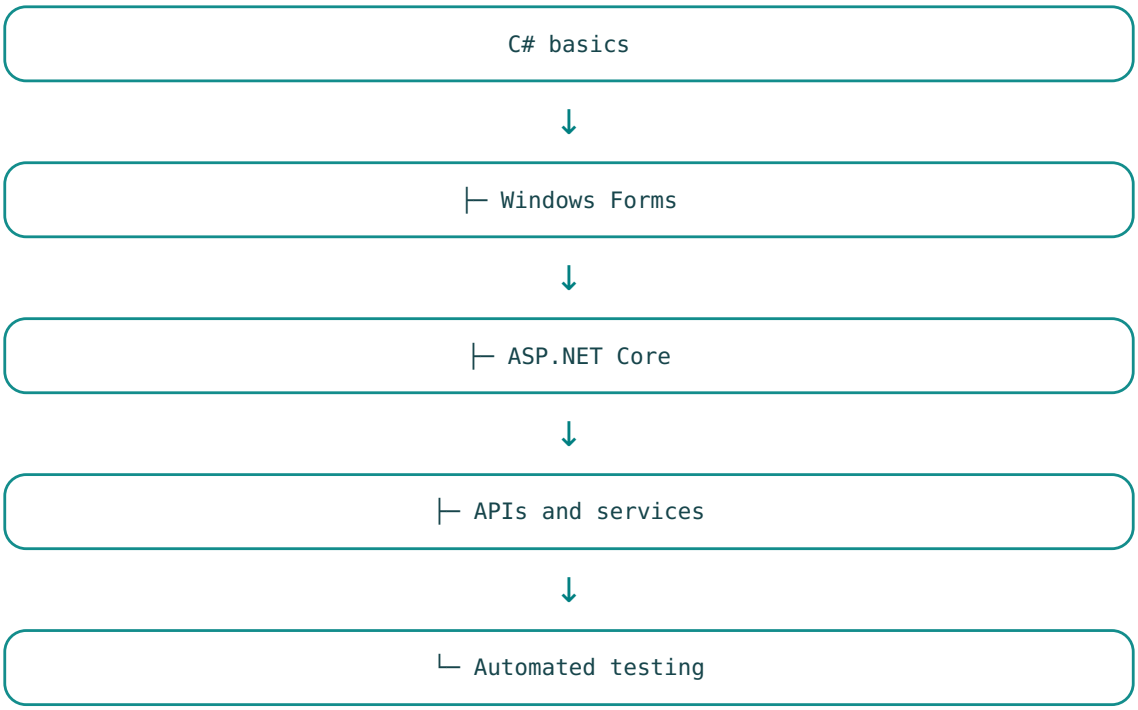
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

එවැනි තොරතුරු සඳහා අවශ්‍ය ආරක්ෂාව නොමැති නිසා.

LESSON 129 / 143

2026 .NET career map

C# සහ .NET මඟින් Console, Windows desktop, ASP.NET Core Web APIs වගේ විවිධ මාර්ග ඉගෙනගන්න පුළුවන්. පළමුව basics + testing හොඳට ඉගෙනගන්න.



ඇයි මෙය වැදගත්?

Framework නම මතක තබාගැනීමට වඩා logic, documentation සහ debugging දක්ෂතා වැදගත්.

පුංචි අභ්‍යාසය

ඊළඟට ඔබ ඉගෙනගන්න කැමති මාර්ගය තෝරන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

ඕනෑම මාර්ගයක්, පැහැදිලි හේතුවක් සමඟ.

LESSON 130 / 143

Lab 01 - Hello Name

මුල් Console app එකට නමක් input කර personalized greeting එකක් පෙන්වන method එකක් එකතු කරන්න.

Windows Command Prompt / CMD

```
Your name: Amal
Hello, Amal!
```

ඇයි මෙය වැදගත්?

User input කියවීම, string, Console.WriteLine නැවත පුහුණු කරන්න.

පුංචි අභ්‍යාසය

"Nila" දුන්නොත් "Hello, Nila!" පෙන්වෙන බව screenshot එකක් ගන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Input වෙනස් වූ විට output හි නමත් වෙනස් විය යුතුයි.

LESSON 131 / 143

Lab 02 - Pocket money

Userගෙන් pocket money value එකක් ලබාගෙන Rs 50 වියදම් කළ පසු ඉතිරිය ගණනය කරන්න.

Windows Command Prompt / CMD

```
Pocket money: 300
Spend: 50
Remaining: Rs 250.00
```

ඇයි මෙය වැදගත්?

Decimal + subtraction + TryParse එකම ලැබ් එකකින් පුහුණු කළ හැකියි.

පුංචි අභ්‍යාසය

Input negative නම් error පෙන්වන condition එකක් එකතු කරන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Positive values පමණක් allow කරන්න.

LESSON 132 / 143

Lab 03 - Category counter

Expenses අතර Food category count එක කියන්න.

Windows Command Prompt / CMD

```
Food: Snack, Mango, Rice
Travel: Bus
Food count = 3
```

ඇයි මෙය වැදගත්?

List, foreach, if condition එකට ප්‍රායෝගික උදාහරණයක්.

පුංචි අභ්‍යාසය

List එකේ Food count වෙනස් කර output බලන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

නිවැරදි count ලැබිය යුතුයි.

LESSON 133 / 143

Lab 04 - Month report

October සහ September expenses මිශ්‍ර කර මාසයක් වෙනුවෙන් total ගණනය කරන්න.

Windows Command Prompt / CMD

```
October: 100 + 250 = 350
September: 70
```

ඇයි මෙය වැදගත්?

DateOnly filtering සහ decimal addition ගැන practice.

පුංචි අභ්‍යාසය

October total 350 ලැබෙන්නේ ඇයි කියලා වාක්‍යයකින් ලියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

September record එක filter වී යන නිසා.

LESSON 134 / 143

Lab 05 - Blank description

Description එක හිස්තැන් පමණක් නම් reject කරවන unit test එකක් ලියන්න.



ඇයි මෙය වැදගත්?

Trim සහ validation එකේ ගුණය තේරුම් ගන්න.

පුංචි අභ්‍යාසය

එම test එකට TC-13 කියලා නමක් දෙන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

TC-13 Blank description rejected.

LESSON 135 / 143

Lab 06 - Add new category

Category list එකට "Games" එකතු කරන්න. UI dropdown හා reports වලට එය නිවැරදිව පෙනෙන බව පරීක්ෂා කරන්න.

```
Before: Food, Travel, School, Other
```

↓

```
After:  Food, Travel, School, Other, Games
```

ඇයි මෙය වැදගත්?

Shared list වෙනස් කළ විට එකම තැනක rule update වීම වාසිදායක.

පුංචි අභ්‍යාසය

අලුත් category එකට අදාළ test case එකක් ලියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Games category එකෙන් add/save/list සාර්ථක.

LESSON 136 / 143

Lab 07 - Export QA

Description එකේ comma සහ double quotation තිබුණත් CSV එක විවෘත කළ විට columns බිඳ නොවැටෙන බව පරීක්ෂා කරන්න.

```
Windows Command Prompt / CMD

Description: Book, "Blue"
CSV field: "Book, ""Blue"""
```

ඇයි මෙය වැදගත්?

Data interchange format සහ boundary testing ගැන practice.

පුංචි අභ්‍යාසය

මෙම string එකට Excel columns ගණන 4ම තිබෙනවාද බලන්න.

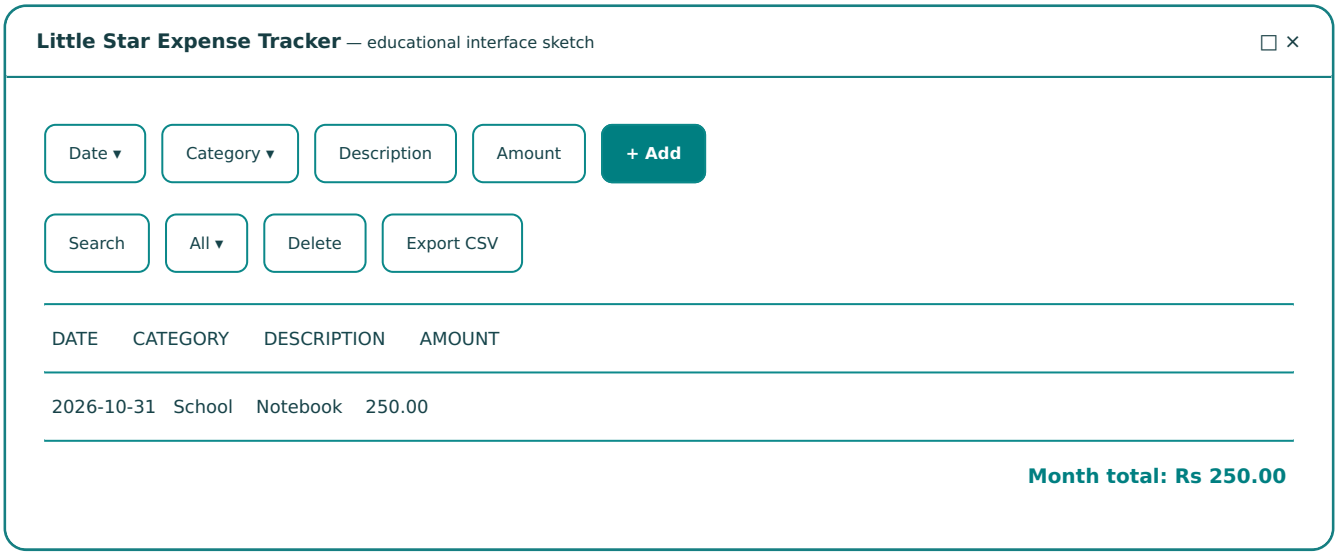
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

හරි quoting තිබේ නම් 4 columns.

LESSON 137 / 143

Lab 08 - UI improvement

Window එකේ buttons අතර ඉඩ, error hints සහ column widths වැඩිදියුණු කරන්න. Testers දෙදෙනෙකුගෙන් feedback ලබාගන්න.



ඇයි මෙය වැදගත්?

Readable interface එකක් user friendly app එකකට අත්‍යවශ්‍යයි.

පුංචි අභ්‍යාසය

ඔබ වෙනස් කළ UI decision 2ක් documentation එකේ සටහන් කරන්න.

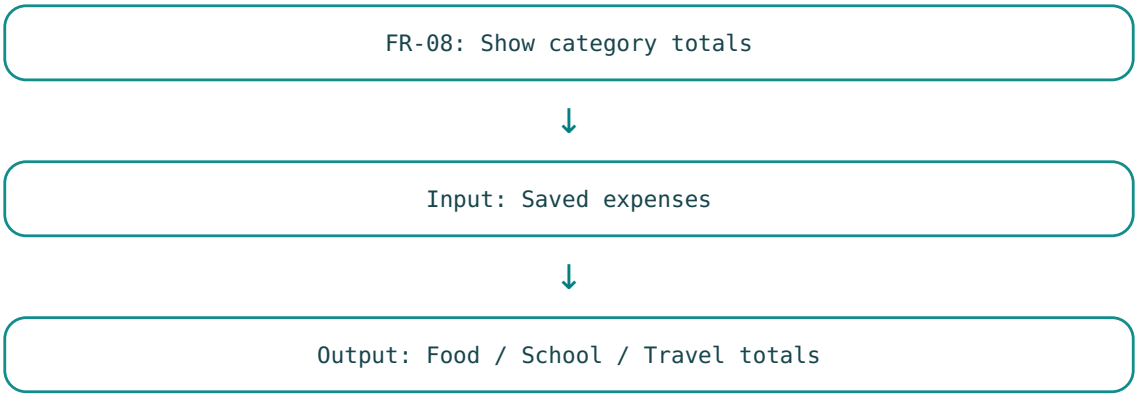
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

උදාහරණ: wider amount column, clearer hint.

LESSON 138 / 143

Lab 09 - New feature plan

Category total chart එකක් අනාගත version එකකට සැලසුම් කරන්න. Requirements සහ wireframe එක ලියන්න; code ලියන්න අවශ්‍ය නැහැ.



ඇයි මෙය වැදගත්?

Coding කිරීමට පෙර scope හඳුනාගැනීමේ practice.

පුංචි අභ්‍යාසය

FR-08 සඳහා acceptance criterion එකක් ලියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Food = 100+200 නම් chart Rs 300 පෙන්විය යුතුයි.

LESSON 139 / 143

Lab 10 - Mini demo

අවසාන දවසේ Console app run කර record add කරන්න, Desktop app open කර list බලන්න, QA checklist පිළිවෙලින් පෙන්වන්න.

Windows Command Prompt / CMD

```
DEMO ORDER:
1 Open app
2 Add Notebook Rs 250
3 Show month total
4 Search and export
5 Show test record
```

ඇයි මෙය වැදගත්?

අනිත් අයෙකුට app එක explain කළ හැකි බවට මෙය practical milestone එකකි.

පුංචි අභ්‍යාසය

Video/demo එකකට විනාඩි 2ක සිංහල narration එකක් සැලසුම් කරන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Problem → solution → demo → testing යන ගමන.

LESSON 140 / 143

Quiz 01 - C# මූලික

විෂය කීපයක් හොඳින් මතක තබාගන්න multiple-choice ප්‍රශ්න තුනකට පිළිතුරු සපයන්න.

```
Windows Command Prompt / CMD

1) Money type? A int B decimal C bool
2) Input? A ReadLine B WriteLine C WriteAllText
3) Loop? A foreach B namespace C class
```

ඇයි මෙය වැදගත්?

Self-check එකෙන් නැවත කියවිය යුතු පාඩම් හඳුනාගන්න.

පුංචි අභ්‍යාසය

ඔබේ පිළිතුරු 1-3 වශයෙන් සටහන් කරන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

1-B, 2-A, 3-A.

LESSON 141 / 143

Quiz 02 - .NET සහ UI

Desktop project එකේ framework සහ important UI controls ගැන check එකක්.

```
Windows Command Prompt / CMD

1) WinForms runs on? A Windows B Browser C Any OS
2) UI click handler? A event B JSON C namespace
3) Show rows? A DataGridView B bool C decimal
```

ඇයි මෙය වැදගත්?

Compiler සහ GUI concepts එකිනෙක සම්බන්ධව මතක තබාගන්න.

පුංචි අභ්‍යාසය

ඔබේ පිළිතුරු 1-3 සටහන් කරන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

1-A, 2-A, 3-A.

LESSON 142 / 143

Quiz 03 - Testing

Quality requirements සහ QA testing ගැන self-check එකක්.

```
Windows Command Prompt / CMD

1) Amount 0? A Accept B Reject
2) Test fix after bug? A Retest B Ignore
3) SRS means? A Requirements B Styling
```

ඇයි මෙය වැදගත්?

Code ලියන තරමට quality පරීක්ෂාවත් වැදගත් බව අවධාරණය කරන්න.

පුංචි අභ්‍යාසය

ඔබේ පිළිතුරු 1-3 සටහන් කරන්න.

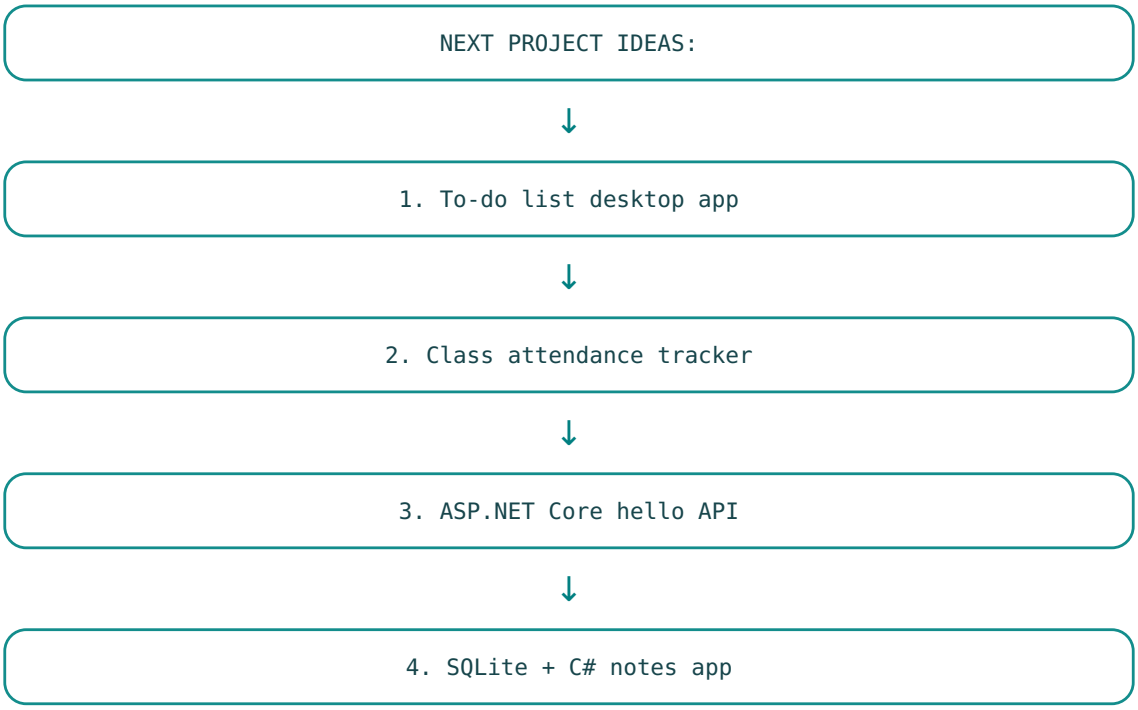
පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

1-B, 2-A, 3-A.

LESSON 143 / 143

ඉදිරි ඉගෙනුම

ඔබට දැන් .NET basics, C# rules, Console සහ Desktop UI, JSON storage, testing ගැන ආරම්භක අවබෝධයක් ඇත. ඊළඟට ASP.NET Core Web API හෝ SQLite database එකක් ඉගෙනගන්න පුළුවන්.



ඇයි මෙය වැදගත්?

වැඩ කරන කුඩා project ගොඩක් හදන එක අන්ධ ලෙස විශාල tutorial එකක් කියවීමට වඩා වටිනවා.

පුංචි අභ්‍යාසය

ඔබේ ඊළඟ project එක තෝරා SRS bullet 3ක් ලියන්න.

පරීක්ෂා කළ යුතු පිළිතුර / ඉඟිය

Problem, two features, one test case.

සම්පූර්ණ Example Project Files

පොතේ පාඩම් සම්බන්ධ කරන්නේ ZIP එකේම source files සමඟයි. පහතින් ප්‍රධාන code files, section-by-section පෙන්වනවා. මෙය source code කියවීමේ අභ්‍යාසයක්; files සම්පූර්ණයෙන් ZIP තුළ ඇත.

PROJECT EXPLORER

```
LittleStarExpenses/
--- Core/Expense.cs
--- Core/ExpenseManager.cs
--- Core/ExpenseStore.cs
--- Console/Program.cs
--- Desktop/MainForm.cs
--- Tests/Program.cs
```

කියවන විදිහ

File එකක code කොටස් අනුපිළිවෙළින් කියවන්න. Typing කරන විට ZIP එකේ `.cs` file එකම open කර copy/edit කරන්න. Appendix extracts තනි standalone programs නොවේ.

Expense.cs - 1-9

Project: LittleStarExpenses.Core/Expense.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Core/Expense.cs • lines 1-9

```
EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
namespace LittleStarExpenses.Core;

// One expense entry in our learning project.
public sealed record Expense(
    Guid Id,
    DateOnly Date,
    string Category,
    string Description,
    decimal Amount);
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

ExpenseManager.cs - 1-18

Project: LittleStarExpenses.Core/ExpenseManager.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Core/ExpenseManager.cs • lines 1-18

```
EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
using System.Globalization;

namespace LittleStarExpenses.Core;

// Rules are kept separate from buttons and file storage.
public sealed class ExpenseManager
{
    public static readonly string[] Categories =
        ["Food", "Travel", "School", "Other"];

    private readonly List<Expense> _expenses = [];

    public IReadOnlyList<Expense> All => _expenses.AsReadOnly();

    public void Load(IEnumerable<Expense> expenses)
    {
        _expenses.Clear();
        foreach (Expense item in expenses)
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

ExpenseManager.cs - 19-36

Project: LittleStarExpenses.Core/ExpenseManager.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Core/ExpenseManager.cs • lines 19-36

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
        {
            if (item.Amount > 0m && item.Amount <= 1_000_000m &&
                Categories.Contains(item.Category) &&
                !string.IsNullOrWhiteSpace(item.Description))
            {
                _expenses.Add(item);
            }
        }
    }

    public Expense Add(DateOnly date, string category,
        string description, decimal amount)
    {
        category = category.Trim();
        description = description.Trim();

        if (!Categories.Contains(category))
            throw new ArgumentException("Choose a valid category.");
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

ExpenseManager.cs - 37-54

Project: LittleStarExpenses.Core/ExpenseManager.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Core/ExpenseManager.cs • lines 37-54

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
        if (description.Length is < 1 or > 80)
            throw new ArgumentException("Description must be 1 to 80
characters.");
        if (amount <= 0m || amount > 1_000_000m)
            throw new ArgumentException("Amount must be more than 0 and at most
1,000,000.");

        Expense item = new(Guid.NewGuid(), date, category, description, amount);
        _expenses.Add(item);
        return item;
    }

    public bool Delete(Guid id)
    {
        Expense? item = _expenses.FirstOrDefault(x => x.Id == id);
        return item is not null && _expenses.Remove(item);
    }

    public IEnumerable<Expense> Find(string query, string category)
    {
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

ExpenseManager.cs - 55-72

Project: LittleStarExpenses.Core/ExpenseManager.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Core/ExpenseManager.cs • lines 55-72

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
        IEnumerable<Expense> result = _expenses;
        if (!string.IsNullOrWhiteSpace(category) && category != "All")
            result = result.Where(x => x.Category == category);
        if (!string.IsNullOrWhiteSpace(query))
            result = result.Where(x => x.Description.Contains(
                query.Trim(), StringComparison.OrdinalIgnoreCase));
        return result.OrderByDescending(x => x.Date);
    }

    public decimal TotalForMonth(int year, int month) =>
        _expenses.Where(x => x.Date.Year == year && x.Date.Month == month)
                 .Sum(x => x.Amount);

    public IReadOnlyDictionary<string, decimal> TotalsByCategory() =>
        _expenses.GroupBy(x => x.Category)
                 .ToDictionary(g => g.Key, g => g.Sum(x => x.Amount));

    // CSV quoting makes commas and double quotes safe inside text fields.
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

ExpenseManager.cs - 73-84

Project: LittleStarExpenses.Core/ExpenseManager.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Core/ExpenseManager.cs • lines 73-84

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
    public string ToCsv()
    {
        static string Quote(string value) => "\"" + value.Replace("\"", "\"\"")
+ "\"";
        List<string> rows = ["Date,Category,Description,Amount"];
        foreach (Expense item in _expenses.OrderBy(x => x.Date))
        {
            rows.Add($"{item.Date:yyyy-MM-dd},{Quote(item.Category)}," +
                     $"{Quote(item.Description)},{item.Amount.ToString("0.00",
CultureInfo.InvariantCulture)}");
        }
        return string.Join(Environment.NewLine, rows);
    }
}
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

ExpenseStore.cs - 1-18

Project: LittleStarExpenses.Core/ExpenseStore.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Core/ExpenseStore.cs • lines 1-18

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
using System.Text.Json;

namespace LittleStarExpenses.Core;

// Local JSON storage: suitable for a beginner project, NOT for banking.
public sealed class ExpenseStore
{
    private readonly string _path;

    public ExpenseStore(string? path = null)
    {
        string folder = Path.Combine(

Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData),
            "LittleStarExpenses");
        _path = path ?? Path.Combine(folder, "expenses.json");
    }

    public string DataPath => _path;
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

ExpenseStore.cs - 19-36

Project: LittleStarExpenses.Core/ExpenseStore.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Core/ExpenseStore.cs • lines 19-36

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
    public List<Expense> Load()
    {
        if (!File.Exists(_path)) return [];
        string json = File.ReadAllText(_path);
        return JsonSerializer.Deserialize<List<Expense>>(json) ?? [];
    }

    public void Save(IEnumerable<Expense> expenses)
    {
        string? folder = Path.GetDirectoryName(_path);
        if (folder is not null) Directory.CreateDirectory(folder);
        string json = JsonSerializer.Serialize(expenses, new
JsonSerializerOptions
        {
            WriteIndented = true
        });
        string temp = _path + ".tmp";
        File.WriteAllText(temp, json);
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

ExpenseStore.cs - 37-39

Project: LittleStarExpenses.Core/ExpenseStore.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Core/ExpenseStore.cs • lines 37-39

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
            File.Move(temp, _path, overwrite: true);
        }
    }
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

Program.cs - 1-18

Project: LittleStarExpenses.Console/Program.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Console/Program.cs • lines 1-18

```
EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
using System.Globalization;
using LittleStarExpenses.Core;

ExpenseStore store = new();
ExpenseManager manager = new();
try
{
    manager.Load(store.Load());
}
catch (Exception ex) when (ex is IOException or System.Text.Json.JsonException)
{
    Console.WriteLine("Could not read saved data: " + ex.Message);
}

while (true)
{
    Console.WriteLine("\n=== Little Star Expense Tracker ===");
    Console.WriteLine("1 Add  2 List  3 Monthly total  4 Export CSV  0 Exit");
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

Program.cs - 19-36

Project: LittleStarExpenses.Console/Program.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Console/Program.cs • lines 19-36

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
    Console.Write("Choose: ");
    string? choice = Console.ReadLine();
    if (choice == "0") break;

    try
    {
        if (choice == "1")
        {
            Console.Write("Date (YYYY-MM-DD, blank = today): ");
            string? dateInput = Console.ReadLine();
            DateOnly date = string.IsNullOrWhiteSpace(dateInput)
                ? DateOnly.FromDateTime(DateTime.Today)
                : DateOnly.ParseExact(dateInput, "yyyy-MM-dd",
CultureInfo.InvariantCulture);
            Console.Write("Category (Food/Travel/School/Other): ");
            string category = Console.ReadLine() ?? "";
            Console.Write("Description: ");
            string description = Console.ReadLine() ?? "";
            Console.Write("Amount (Rs): ");
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

Program.cs - 37-54

Project: LittleStarExpenses.Console/Program.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Console/Program.cs • lines 37-54

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
            string amountInput = Console.ReadLine() ?? "";
            if (!decimal.TryParse(amountInput, NumberStyles.Number,
                CultureInfo.CurrentCulture, out decimal amount))
                throw new ArgumentException("Enter a number such as 250.50.");
            Expense item = manager.Add(date, category, description, amount);
            store.Save(manager.All);
            Console.WriteLine("Saved: " + item.Description);
        }
        else if (choice == "2")
        {
            foreach (Expense item in manager.Find("", "All"))
                Console.WriteLine($"{item.Date:yyyy-MM-dd} | {item.Category,-7}
| " +
                    $"{item.Description,-18} | Rs {item.Amount:0.00}");
        }
        else if (choice == "3")
        {
            Console.WriteLine($"This month's total: Rs " +
                manager.TotalForMonth(DateTime.Today.Year,
DateTime.Today.Month).ToString("0.00"));
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

Program.cs - 55-70

Project: LittleStarExpenses.Console/Program.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Console/Program.cs • lines 55-70

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
        }
        else if (choice == "4")
        {
            string path = Path.Combine(Environment.GetFolderPath(
                Environment.SpecialFolder.DesktopDirectory), "little-star-
expenses.csv");
            File.WriteAllText(path, manager.ToCsv());
            Console.WriteLine("Exported: " + path);
        }
        else Console.WriteLine("Choose 0, 1, 2, 3 or 4.");
    }
    catch (Exception ex) when (ex is ArgumentException or FormatException or
IOException
        or UnauthorizedAccessException)
    {
        Console.WriteLine("Please fix: " + ex.Message);
    }
}
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

Program.cs - 1-11

Project: LittleStarExpenses.Desktop/Program.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Desktop/Program.cs • lines 1-11

```
EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/
```

```
namespace LittleStarExpenses.Desktop;

internal static class Program
{
    [STAThread]
    static void Main()
    {
        ApplicationConfiguration.Initialize();
        Application.Run(new MainForm());
    }
}
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

MainForm.cs - 1-18

Project: LittleStarExpenses.Desktop/MainForm.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Desktop/MainForm.cs • lines 1-18

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
using System.Globalization;
using LittleStarExpenses.Core;

namespace LittleStarExpenses.Desktop;

// Designer-free Windows Forms example: every control is created in C#.
public sealed class MainForm : Form
{
    private readonly ExpenseManager manager = new();
    private readonly ExpenseStore store = new();
    private readonly DateTimePicker date = new() { Format =
DateTimePickerFormat.Short, Width = 140 };
    private readonly ComboBox category = new() { DropDownStyle =
ComboBoxStyle.DropDownList, Width = 125 };
    private readonly TextBox description = new() { Width = 220, PlaceholderText
= "e.g. Notebook" };
    private readonly TextBox amount = new() { Width = 100, PlaceholderText =
"250.00" };
    private readonly TextBox search = new() { Width = 180, PlaceholderText =
"Search description" };
    private readonly ComboBox filter = new() { DropDownStyle =
ComboBoxStyle.DropDownList, Width = 115 };
    private readonly Label totalLabel = new() { AutoSize = true, Font = new
Font("Segoe UI", 12, FontStyle.Bold) };
    private readonly DataGridView table = new()
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

MainForm.cs - 19-36

Project: LittleStarExpenses.Desktop/MainForm.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Desktop/MainForm.cs • lines 19-36

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
{
    Dock = DockStyle.Fill,
    ReadOnly = true,
    AllowUserToAddRows = false,
    AutoGenerateColumns = false,
    SelectionMode = DataGridViewSelectionMode.FullRowSelect,
    MultiSelect = false,
    BackgroundColor = Color.White,
    AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill
};

public MainForm()
{
    Text = "Little Star Expense Tracker - Learning App";
    StartPosition = FormStartPosition.CenterScreen;
    MinimumSize = new Size(820, 520);
    Size = new Size(1020, 620);
    Font = new Font("Segoe UI", 10);
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

MainForm.cs - 37-54

Project: LittleStarExpenses.Desktop/MainForm.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Desktop/MainForm.cs • lines 37-54

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
        BackColor = Color.White;

category.Items.AddRange(ExpenseManager.Categories.Cast<object>().ToArray());
        category.SelectedIndex = 0;
        filter.Items.Add("All");

filter.Items.AddRange(ExpenseManager.Categories.Cast<object>().ToArray());
        filter.SelectedIndex = 0;

        FlowLayoutPanel inputBar = new()
        {
            Dock = DockStyle.Top, AutoSize = true, Padding = new Padding(12),
            WrapContents = true, BackColor = Color.White
        };
        inputBar.Controls.AddRange([
            new Label { Text = "Date", AutoSize = true, Margin = new Padding(5,
11, 6, 5) },
            date,
            new Label { Text = "Category", AutoSize = true, Margin = new
Padding(10, 11, 6, 5) },
            category,
            description,
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

MainForm.cs - 55-72

Project: LittleStarExpenses.Desktop/MainForm.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Desktop/MainForm.cs • lines 55-72

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
            amount
        ]);
        Button addButton = CreateButton("+ Add expense");
        addButton.Click += (_, _) => AddExpense();
        inputBar.Controls.Add(addButton);

        FlowLayoutPanel toolsBar = new()
        {
            Dock = DockStyle.Top, AutoSize = true, Padding = new Padding(12),
            WrapContents = true, BackColor = Color.White
        };
        Button deleteButton = CreateButton("Delete selected");
        deleteButton.Click += (_, _) => DeleteSelected();
        Button exportButton = CreateButton("Export CSV");
        exportButton.Click += (_, _) => ExportCsv();
        toolsBar.Controls.AddRange([search, filter, deleteButton, exportButton,
totalLabel]);
        search.TextChanged += (_, _) => RefreshTable();
        filter.SelectedIndexChanged += (_, _) => RefreshTable();
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

MainForm.cs - 73-90

Project: LittleStarExpenses.Desktop/MainForm.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Desktop/MainForm.cs • lines 73-90

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
        table.Columns.Add(new DataGridViewTextBoxColumn { Name = "Date",
HeaderText = "Date", DataPropertyName = "Date" });
        table.Columns.Add(new DataGridViewTextBoxColumn { Name = "Category",
HeaderText = "Category", DataPropertyName = "Category" });
        table.Columns.Add(new DataGridViewTextBoxColumn { Name = "Description",
HeaderText = "Description", DataPropertyName = "Description" });
        table.Columns.Add(new DataGridViewTextBoxColumn { Name = "Amount",
HeaderText = "Amount (Rs)", DataPropertyName = "Amount" });
        table.Columns.Add(new DataGridViewTextBoxColumn { Name = "Id", Visible =
false, DataPropertyName = "Id" });

        Controls.Add(table);
        Controls.Add(toolsBar);
        Controls.Add(inputBar);

        try { manager.Load(store.Load()); }
        catch (Exception ex) when (ex is IOException or
System.Text.Json.JsonException)
        { MessageBox.Show("Saved data cannot be opened: " + ex.Message, "Data
warning"); }
        RefreshTable();
    }

    private static Button CreateButton(string text) => new()
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

MainForm.cs - 91-108

Project: LittleStarExpenses.Desktop/MainForm.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Desktop/MainForm.cs • lines 91-108

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
    {
        Text = text, AutoSize = true, Height = 34, Margin = new Padding(7, 2, 5,
2)
    };

    private void AddExpense()
    {
        if (!decimal.TryParse(amount.Text.Trim(), NumberStyles.Number,
            CultureInfo.CurrentCulture, out decimal value))
        {
            MessageBox.Show("Enter a valid amount, such as 250.50.");
            return;
        }
        try
        {
            manager.Add(DateOnly.FromDateTime(date.Value),
                category.Text, description.Text, value);
            store.Save(manager.All);
            description.Clear(); amount.Clear();
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

MainForm.cs - 109-126

Project: LittleStarExpenses.Desktop/MainForm.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Desktop/MainForm.cs • lines 109-126

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
            RefreshTable();
        }
        catch (Exception ex) when (ex is ArgumentException or IOException or
UnauthorizedAccessException)
        { MessageBox.Show(ex.Message, "Please check your entry"); }
    }

    private void DeleteSelected()
    {
        if (table.CurrentRow?.Cells["Id"].Value is not Guid id) return;
        if (MessageBox.Show("Delete this sample record?", "Confirm",
            MessageBoxButtons.YesNo) != DialogResult.Yes) return;
        try { manager.Delete(id); store.Save(manager.All); RefreshTable(); }
        catch (IOException ex) { MessageBox.Show("Save failed: " +
ex.Message); }
    }

    private void ExportCsv()
    {
        using SaveFileDialog dialog = new()
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

MainForm.cs - 127-144

Project: LittleStarExpenses.Desktop/MainForm.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Desktop/MainForm.cs • lines 127-144

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
        {
            FileName = "little-star-expenses.csv", Filter = "CSV file (*.csv)|
*.csv"
        };
        if (dialog.ShowDialog() != DialogResult.OK) return;
        try
        {
            File.WriteAllText(dialog.FileName, manager.ToCsv());
            MessageBox.Show("CSV exported!");
        }
        catch (Exception ex) when (ex is IOException or
UnauthorizedAccessException)
        { MessageBox.Show("Export failed: " + ex.Message); }
    }

    private void RefreshTable()
    {
        table.DataSource = manager.Find(search.Text, filter.Text).Select(x =>
new
        {
            Date = x.Date.ToString("yyyy-MM-dd"),
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

MainForm.cs - 145-150

Project: LittleStarExpenses.Desktop/MainForm.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Desktop/MainForm.cs • lines 145-150

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
            x.Category, x.Description, Amount = x.Amount.ToString("0.00"), x.Id
        }).ToList();
        totalLabel.Text = "  Month total: Rs " +
            manager.TotalForMonth(DateTime.Today.Year,
DateTime.Today.Month).ToString("0.00");
    }
}
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

Program.cs - 1-18

Project: LittleStarExpenses.Tests/Program.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Tests/Program.cs • lines 1-18

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
using LittleStarExpenses.Core;

// A dependency-free test runner for young beginners.
int passed = 0;
int failed = 0;
void Test(string name, Action action)
{
    try { action(); Console.WriteLine("PASS: " + name); passed++; }
    catch (Exception ex) { Console.WriteLine("FAIL: " + name + " -> " +
ex.Message); failed++; }
}
void Check(bool condition, string message)
{
    if (!condition) throw new Exception(message);
}

Test("New list starts empty", () => Check(new ExpenseManager().All.Count == 0,
"not empty"));
Test("Adding expense changes count", () =>
{
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

Program.cs - 19-36

Project: LittleStarExpenses.Tests/Program.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Tests/Program.cs • lines 19-36

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
    ExpenseManager m = new(); m.Add(new DateOnly(2026, 10, 1), "School", "Pen",
75m);
    Check(m.All.Count == 1, "not added");
});
Test("Zero amount rejected", () =>
{
    try { new ExpenseManager().Add(new DateOnly(2026, 10, 1), "School", "Pen",
0m); }
    catch (ArgumentException) { return; }
    throw new Exception("zero accepted");
});
Test("Invalid category rejected", () =>
{
    try { new ExpenseManager().Add(new DateOnly(2026, 10, 1), "Hacks", "Pen",
10m); }
    catch (ArgumentException) { return; }
    throw new Exception("category accepted");
});
Test("Monthly total excludes other months", () =>
{
    ExpenseManager m = new();
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

Program.cs - 37-54

Project: LittleStarExpenses.Tests/Program.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Tests/Program.cs • lines 37-54

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
    m.Add(new DateOnly(2026, 10, 1), "Food", "Snack", 100m);
    m.Add(new DateOnly(2026, 9, 1), "Food", "Snack", 30m);
    Check(m.TotalForMonth(2026, 10) == 100m, "wrong total");
});
Test("Deleting an expense removes it", () =>
{
    ExpenseManager m = new();
    Expense e = m.Add(new DateOnly(2026, 10, 1), "Travel", "Bus", 90m);
    Check(m.Delete(e.Id) && m.All.Count == 0, "delete failed");
});
Test("Search is case insensitive", () =>
{
    ExpenseManager m = new();
    m.Add(new DateOnly(2026, 10, 1), "School", "NOTEBOOK", 100m);
    Check(m.Find("notebook", "All").Count() == 1, "search failed");
});
Test("CSV quotes a comma", () =>
{
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

Program.cs - 55-72

Project: LittleStarExpenses.Tests/Program.cs. සම්පූර්ණ file එක ZIP එකේ අඩංගුයි.

Visual Studio Code • LittleStarExpenses.Tests/Program.cs • lines 55-72

EXPLORER
--- Core/
--- Console/
--- Desktop/
--- Tests/

```
    ExpenseManager m = new();
    m.Add(new DateOnly(2026, 10, 1), "Other", "Book, blue", 100m);
    Check(m.ToCsv().Contains("\"Book, blue\""), "CSV escaping failed");
});
Test("JSON round-trip", () =>
{
    string p = Path.Combine(Path.GetTempPath(), "egotech-test-" + Guid.NewGuid()
+ ".json");
    try
    {
        ExpenseStore s = new(p);
        ExpenseManager m = new(); m.Add(new DateOnly(2026, 10, 1), "Food",
"Mango", 55m);
        s.Save(m.All);
        Check(s.Load().Count == 1, "load failed");
    }
    finally { if (File.Exists(p)) File.Delete(p); }
});
Console.WriteLine($"\nTests passed: {passed}; failed: {failed}");
return failed == 0 ? 0 : 1;
```

Code reading challenge

මෙම section එකේ method එකක්, variable එකක් සහ comment එකක් හඳුනාගන්න. ඉන්පසු original `.cs` file එකේ ඒ කොටස සොයා බලන්න.

SRS - Project Requirements

මෙම පිටුවේ සටහන් භාවිතා කර තමන්ගේම project notebook එකක් සකස් කරගන්න.

Functional requirements

Add, list, search, filter, delete after confirmation, month total, CSV export and local save.

Nonfunctional requirements

Offline app; clear input errors; readable Windows UI; no sensitive financial information.

Acceptance criterion example

Given amount = 0; When user clicks Add; Then show an error and do not save.

QA - Sample Test Matrix

මෙම පිටුවේ සටහන් භාවිතා කර තමන්ගේම project notebook එකක් සකස් කරගන්න.

TC-01

Add School / Notebook / Rs 250: new row appears and persists.

TC-02

Amount 0: reject without adding.

TC-03

Description spaces only: reject.

TC-04

Save, exit, reopen: existing records remain.

TC-05

Export CSV with a comma inside description: Excel columns stay correct.

7-Day Practice Plan

මෙම පිටුවේ සටහන් භාවිතා කර තමන්ගේම project notebook එකක් සකස් කරගන්න.

Day 1 & 2

Install .NET 10 and Visual Studio; run Console Hello World; practice variables, conditions.

Day 3 & 4

Run Core/Console; create expenses; learn JSON and monthly total.

Day 5

Run Windows Forms UI; test search, filter and export.

Day 6

Run tests; execute manual QA cases; write one bug report.

Day 7

Record a demo; prepare SRS, test report and explain limitations.

Common Errors - Troubleshooting

මෙම පිටුවේ සටහන් භාවිතා කර තමන්ගේම project notebook එකක් සකස් කරගන්න.

dotnet not recognized

Install SDK; reopen CMD; test `dotnet --version`.

No Windows Forms template

Check Visual Studio .NET desktop development workload; make sure on Windows.

Amount rejected

Use a valid positive amount; decimal separators follow Windows culture.

Data not saved

Check permissions for LocalApplicationData; do not modify arbitrary protected folders.

WinForms not running on Mac/Linux

Windows Forms is Windows-specific; use Console project on other platforms.

Teacher and Parent Notes

මෙම පිටුවේ සටහන් භාවිතා කර තමන්ගේම project notebook එකක් සකස් කරගන්න.

Suggested ages

Age 10+ with adult guidance for software installation and online accounts.

Teaching pace

Teach one concept, demo one example, then one tiny exercise per session.

Fair assessment

Ask learner to explain each statement; do not grade by typing speed.

Safety

Only fabricated money amounts, no bank details or passwords, no claims of financial compliance.

Official Resources

මෙම පිටුවේ සටහන් භාවිතා කර තමන්ගේම project notebook එකක් සකස් කරගන්න.

.NET 10 download

https://dotnet.microsoft.com/en-us/download/dotnet/10.0

Version support policy

https://dotnet.microsoft.com/en-us/platform/support/policy/dotnet-core

C# language guide

https://learn.microsoft.com/en-us/dotnet/csharp/

Windows Forms learning

https://learn.microsoft.com/en-us/dotnet/desktop/winforms/

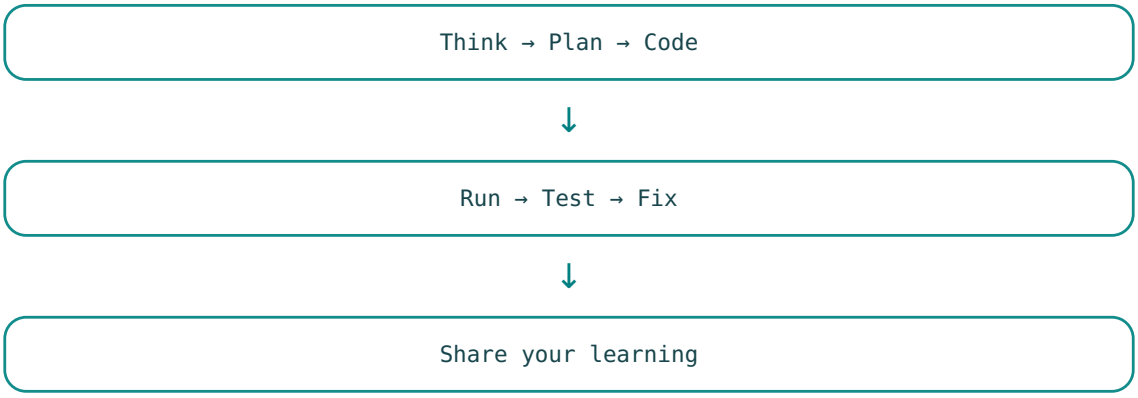
Visual Studio download

https://visualstudio.microsoft.com/

CONGRATULATIONS

දැන් ඔබ C# developer කෙනෙක් වීමට පළමු පියවර ගත්තා!

ඔබ .NET platform එක, C# fundamentals, console සහ desktop UI, data storage, SDLC සහ QA ගැන ආරම්භක දැනුම ලබාගෙන තිබෙනවා.



අවසන් පුංචි පැවරුම

Little Star Expense Tracker app එක run කර කල්පිත records 3ක් ඇතුළත් කරන්න. Screenshot එකක්, test results සටහනක් සහ feature එකක කෙටි විස්තරයක් සකස් කරන්න.

ඊළඟ පාඩමට ASP.NET Core Web API හෝ C# + SQLite Database project එකක් උත්සාහ කරන්න. සතුටින් ඉගෙනගන්න!