ERP SYSTEM BASICS
ERP එකක් කියන්නේ මොකක්ද?

මොකටද? වැඩ කරන්නේ කොහොමද?

Little Star ERP - සරල, run කළ හැකි example project එකක්



One shop · One database · Connected modules

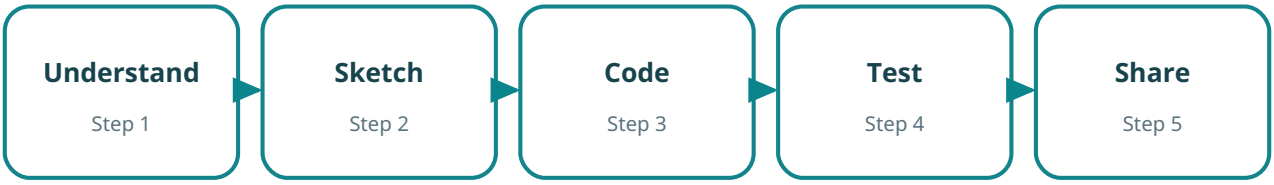
Python · SQLite · CMD · Chrome · SDLC · SRS · Testing & QA
100% සුදු පිටු · කළු Code Panels · egotechworld.com

මේ පොතෙන් ඉගෙනගන්නා ගමන

අද සිට ඔබ trainee business analyst, designer, developer සහ tester කෙනෙක්!

දැනගත යුතු කරුණු

- 01. ERP ඇයි? → 02. Module හා business flow → 03. SDLC/SRS.
- 04. Python + SQLite setup → 05. Mini ERP software build.
- 06. Testing/QA → 07. 2026 ERP trends → 08. MCQ සහ mini challenges.



✎ **ඔබේ අභ්‍යාසය**

ඔබට වැඩිපුරම ඉගෙනගැනීමට ආස කොටස ලියා තබාගන්න.

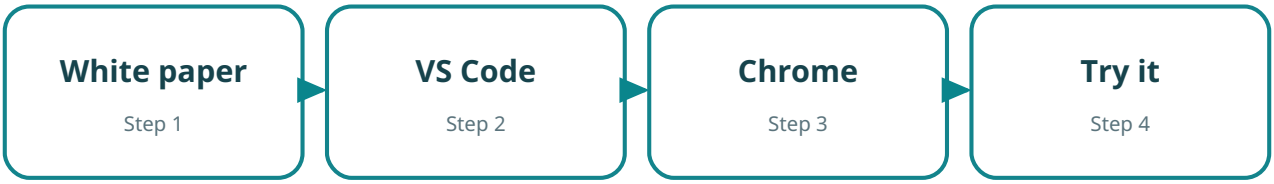
මතක තබාගන්න: Code එක ඇත්තටම run කරලා පරීක්ෂා කරන එක කියවීමට වඩා වැදගත්.

පොතේ විශේෂ design එක

පිටු පසුබිම සම්පූර්ණ සුදුයි. Code panels පමණක් කළුයි. පින්තූර සහ diagrams සරල, පිරිසිදු සහ නිරවුල්.

දැනගත යුතු කරුණු

- සියලුම code snippets VS Code-style dark panels තුළ පෙන්වයි.
- CMD commands වෙනම කළු terminal එකක පෙන්වයි.
- Web previews Chrome-style browser frame එකක පෙන්වයි; සැබෑ project source ZIP එකත් ඇත.



✎ **ඔබේ අභ්‍යාසය**

ඔබේ notebook එකේ “INPUT → PROCESS → OUTPUT” ඇඳන්න.

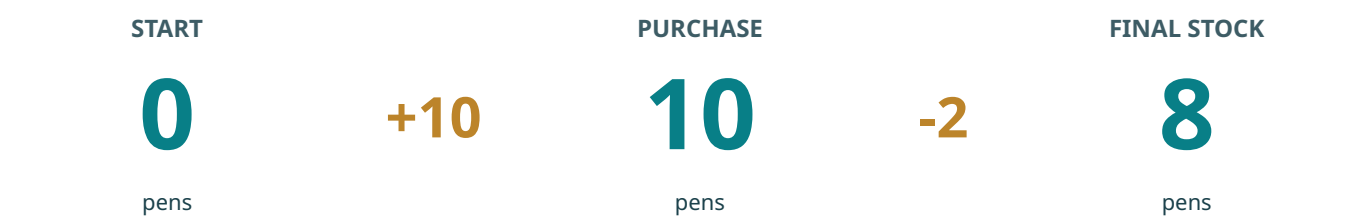
මතක තබාගන්න: Browser previews පැහැදිලි කිරීම සඳහා සෑදූ illustrative views විය හැක.

පළමු sample shop එක

Little Star Shop එකේ Blue Pen, Notebook, Pencil සහ Eraser විකුණනවා කියලා හිතමු.

දැනගත යුතු කරුණු

- Blue Pen එකක selling price Rs 80.00.
- අපි supplier කෙනෙකුගෙන් pen 10ක් ගෙන stock එක වැඩි කරමු.
- පසුව customer කෙනෙකුට pen 2ක් විකුණමු. Stock = 8; sales revenue = Rs 160.00.



✎ ඔබේ අභ්‍යාසය

10 - 2 = 8 සහ 2 × 80 = 160 ඔබම ගණනය කරන්න.

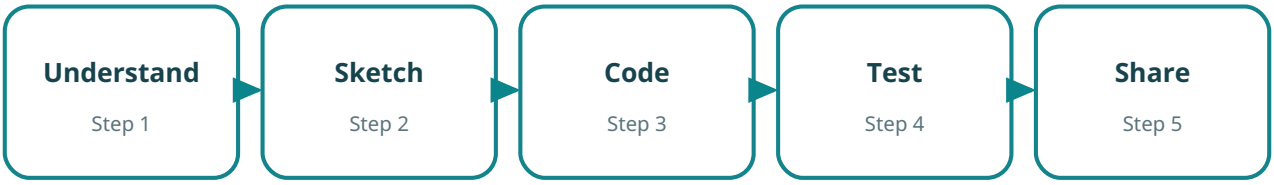
මතක තබාගන්න: Revenue කියන්නේ Profit නොවේ; Cost of goods sold සහ වෙනත් වියදම් වෙනම සලකා බැලිය යුතුයි.

පොත භාවිතා කරන ආකාරය

එක් පිටුවක් = එක් පොඩි idea එකක්. තේරුම් ගන්න, අත්හදා බලන්න, ලියා තබාගන්න.

දැනගත යුතු කරුණු

- පළමුව ප්‍රශ්නය කියවන්න. ඉන්පසු diagram එක බලන්න.
- Code line එක තේරුණාම VS Code එකේ type කර run කරන්න.
- වැරදුණත් කමක් නැහැ — error message එක කියවා නැවත පරීක්ෂා කරන්න.



✎ ඔබේ අභ්‍යාසය

“මට දැනට ERP කියන්නේ ...” කියලා වාක්‍යයක් ලියන්න.

මතක තබාගන්න: Real customer නම්, phone numbers හෝ payment details sample project එකට ඇතුළත් නොකරන්න.

ERP කියන අකුරු තුන

ERP = Enterprise Resource Planning. ලොකු වචන තුනක් වුණත් අදහස ඉතා සරලයි.

දැනගත යුතු කරුණු

- Enterprise = ව්‍යාපාරයක් හෝ සංවිධානයක්.
- Resource = භාණ්ඩ, මුදල්, කාලය, පුද්ගලයන් සහ තොරතුරු.
- Planning = මේ සියල්ල නිවැරදිව සැලසුම් කර සම්බන්ධ කර වැඩ කිරීම.



✎ ඔබේ අභ්‍යාසය

E, R, P සඳහා සිංහල අරුත් තුන ලියන්න.

මතක තබාගන්න: ERP කියන්නේ invoice එකක් print කරන app එකක් පමණක් නොවේ.

එකට සම්බන්ධ වීම ඇයි වැදගත්?

වෙළඳසැලක Sales එකට තනි ගණන් පොතක්, Store room එකට තවත් පොතක් තිබුණොත් ගණන් වෙනස් වෙන්න පුළුවන්.

දැනගත යුතු කරුණු

- Sales එකේ pen 2ක් විකුණූ විට stock අතින් අඩු කරන එක අමතක විය හැක.
- ERP එකේ දෙකම එක වැඩකට සම්බන්ධ කරගන්නවා.
- එවිට manager ට එකම නිවැරදි තොරතුර ඉක්මනින් ලැබේ.

WITHOUT ERP	WITH ERP
Sales notebook: 8 pens	Shared products table: 8
Store notebook: 10 pens	All modules read the same stock
Which one is correct?	Changes recorded together

✎ ඔබේ අභ්‍යාසය

තනි තනි exercise books තුනක් පාවිච්චි කළොත් ඇතිවන ගැටලු දෙකක් ලියන්න.

මතක තබාගන්න: “One source of truth” = එකම විශ්වාස කළ හැකි තොරතුරු මූලාශ්‍රය.

ERP නැත්නම් වෙන්නේ මොකක්ද?

අපේ shop එකේ department තුනකට වෙන වෙනම notebooks තිබෙනවා කියලා හිතන්න.

දැනගත යුතු කරුණු

- Store keeper කියනවා stock = 10 කියලා.
- Cashier අලුත් sale 2ක් කළ නිසා ඇත්තටම stock = 8.
- Manager ට ලැබෙන්නේ පරණ data එකක් නම් වැරදි purchase decision එකක් ගන්න පුළුවන්.

WITHOUT ERP

Sales notebook: 8 pens

Store notebook: 10 pens

Which one is correct?

WITH ERP

Shared products table: 8

All modules read the same stock

Changes recorded together

✎ ඔබේ අභ්‍යාසය

Stock එක 10 ද 8 ද? ඒ වෙනස ඇති වූයේ ඇයි?

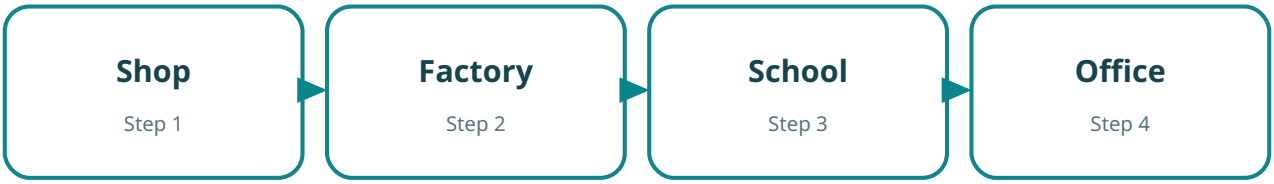
මතක තබාගන්න: ERP තුළත් data වැරදි ඇතුළත් කළොත් වැරදි report ලැබිය හැක.

ERP එකක් භාවිතා කරන්නේ කවුද?

සුපිරි වෙළඳසැල, bakery, hospital, workshop, factory සහ school වැනි තැන්වල ERP සැලසුම් උපකාරී වේ.

දැනගත යුතු කරුණු

- Shop: stock, purchase, sales, reports.
- Factory: raw material, production, distribution, finance.
- School: registrations, fees, supplies, planning (විවිධ systems භාවිතා කළ හැක).



✎ ඔබේ අභ්‍යාසය

ඔබ දන්නා ආයතන තුනක ERP වලින් සම්බන්ධ කළ හැකි වැඩ ලියන්න.

මතක තබාගන්න: ප්‍රමාණය කුඩා වුවත් integration වල වටිනාකම අඩු නොවේ.

ERP එකේ ප්‍රධාන කොටස්

එක විශාල app එක ඇතුළේ ලොකු කාර්යයන් සඳහා වෙන වෙනම modules තිබෙනවා.

දැනගත යුතු කරුණු

- Inventory: භාණ්ඩ stock.
- Purchasing: supplier ගෙන් භාණ්ඩ ගැනීම.
- Sales: customer කෙනෙකුට විකිණීම. Reports: එකම data වලින් සාරාංශ.



✎ ඔබේ අභ්‍යාසය

මෙම modules හතරෙන් stock වෙනස් කරන modules දෙක නම් කරන්න.

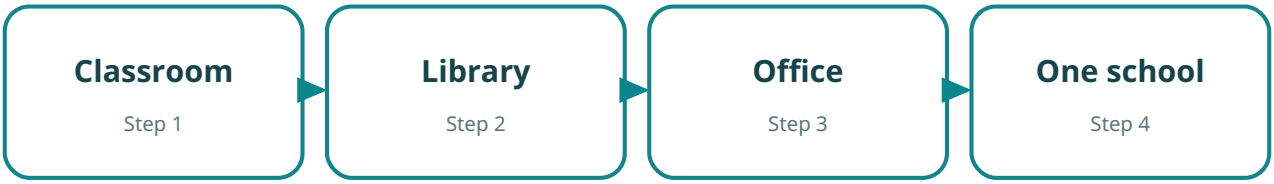
මතක තබාගන්න: Real ERP එකක Finance, HR, Manufacturing, CRM වැනි අමතර modules තිබිය හැක.

Module කියන්නේ කාමරයක් වගේ

පාසලේ library, classroom සහ office වෙන වෙනම කාර්යයන් කරනවා. ඒත් එක පාසලේම කොටස්.

දැනගත යුතු කරුණු

- ERP Inventory module එක භාණ්ඩ ගැන දන්නවා.
- Sales module එක විකුණුම් transaction එක ගැන දන්නවා.
- Shared database එකෙන් module දෙක එකට තොරතුරු හුවමාරු කරයි.



✎ ඔබේ අභ්‍යාසය

ඔබේ පාසලේ විවිධ කාමර තුනක් ERP modules තුනකට සසඳන්න.

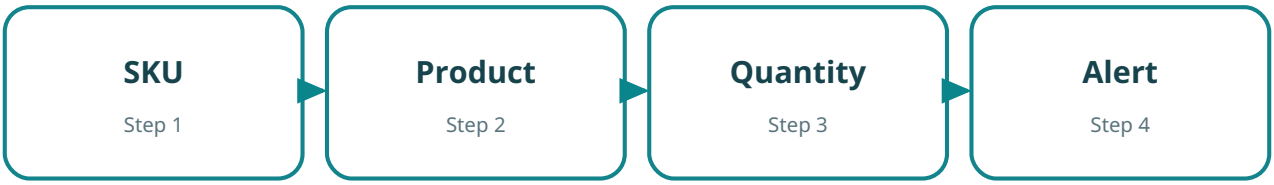
මතක තබාගන්න: Module නම් වෙනස් වුණත් data duplication අඩු වීම අත්‍යවශ්‍යයි.

Inventory කියන්නේ මොකක්ද?

වෙළඳසැලේ විකිණීමට ඇති භාණ්ඩ ගණන සහ පිහිටීම දැන සිටීම.

දැනගත යුතු කරුණු

- SKU = එක් භාණ්ඩ වර්ගයක් හඳුනාගන්න unique code.
- Stock quantity = දැනට විකිණීමට ඇති unit ගණන.
- Reorder level = අඩු වී ඇතිදැයි දැනගන්න තෝරාගත් පරතරය.



✎ ඔබේ අභ්‍යාසය

Blue Pen SKU එක PEN-001 නම් Notebook සඳහා ඔබේ SKU එකක් යෝජනා කරන්න.

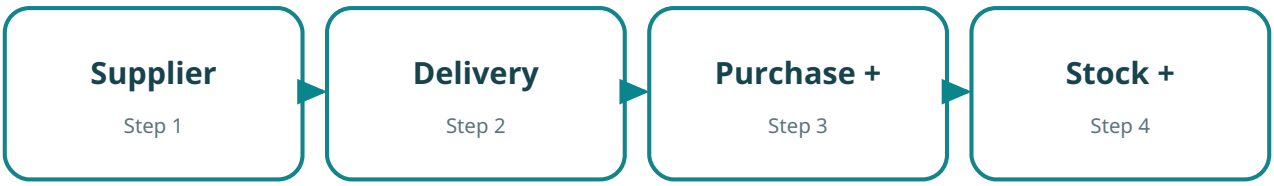
මතක තබාගන්න: Stock = 0 නම් අලුත් sale එක නවත්වන්න ඕන.

Purchasing කියන්නේ මොකක්ද?

Supplier කෙනෙකුගෙන් ව්‍යාපාරයට අලුත් භාණ්ඩ ගෙන ඒමයි.

දැනගත යුතු කරුණු

- Purchase record: product, quantity, unit cost, supplier.
- භාණ්ඩ ඇත්තටම ලැබුණු විට stock quantity වැඩි කරයි.
- Purchased cost සහ selling price දෙක එකම දේ නොවේ.



✎ ඔබේ අභ්‍යාසය

Pen 10ක්, each Rs 50 නම් purchase spend කීයද?

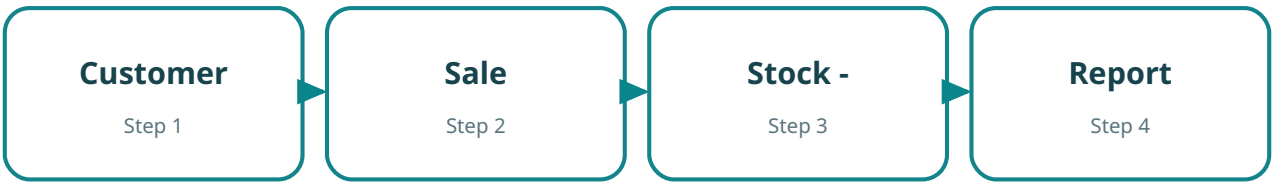
මතක තබාගන්න: Purchase order එකක් සහ goods receipt එකක් real ERP වල වෙන වෙනම stages විය හැක.

Sales කියන්නේ මොකක්ද?

Customer කෙනෙකුට භාණ්ඩ විකුණන වැඩේ system එකේ record කිරීමයි.

දැනගත යුතු කරුණු

- Sale record: product, quantity, captured selling price, total.
- Stock ඇති නම් sale එක save කර quantity අඩු කරයි.
- Stock නැත්නම් helpful error message ලබා දෙයි.



✎ **ඔබේ අභ්‍යාසය**

Pen 2ක් × Rs 80 ක revenue කීයද?

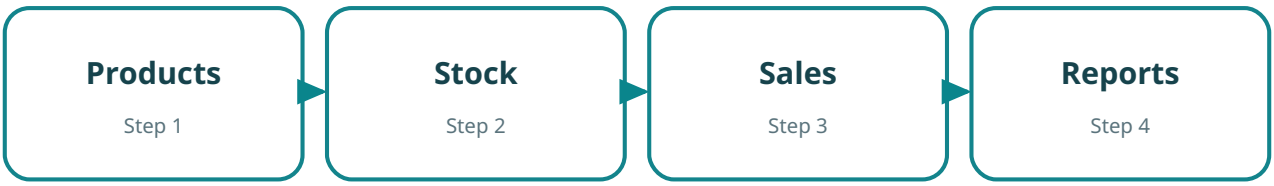
මතක තබාගන්න: Cash payment confirmation වැනි අංග අපේ mini ERP එකේ නැහැ.

Reports මොකටද?

Data එක තේරුම් ගන්න පහසු සංඛ්‍යා සහ වගු බවට පත් කරන එක reporting.

දැනගත යුතු කරුණු

- Total products, units in stock, low-stock items වගේ KPIs.
- Sales revenue සහ purchase spend වෙන වෙනම පෙන්වයි.
- Stock movement history මඟින් කිසිදු වෙනසක් අනුමාන නොකර trace කරන්න පුළුවන්.



✎ ඔබේ අභ්‍යාසය

Manager කෙනෙකුට වැදගත් වාර්තා තුනක් තෝරන්න.

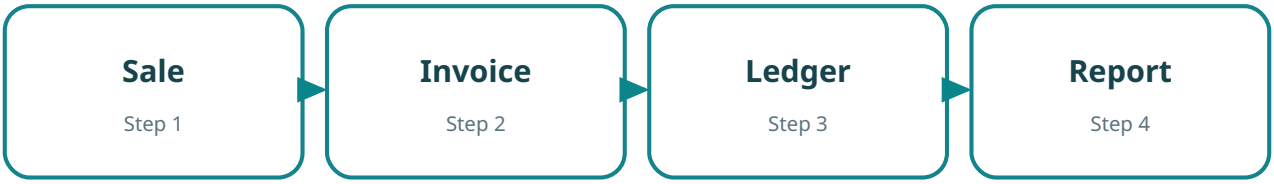
මතක තබාගන්න: Report එකේ “Rs 160 revenue” කියලා තිබුණාට Rs 160 profit කියලා ලියන්න එපා.

Finance module එක වැඩි දියුණු වෙන්නේ කොහොමද?

Real ERP එකක Finance module එකට ගණකාධිකරණයේ සම්මත ක්‍රියාදාම අවශ්‍යයි.

දැනගත යුතු කරුණු

- Invoices, receipts, payables, receivables, journal entries වැනි අංග සලකන්න.
- Transactions correct ledgers වලට ගමන් කළ යුතුයි.
- අපේ school demo reports Finance module එකකට සමාන නොවේ.



✎ ඔබේ අභ්‍යාසය

Revenue, cash සහ profit අතර ඇති වෙනස මතක සටහනක ලියන්න.

මතක තබාගන්න: Tax, VAT, payroll සහ accounting rules real system වලට professional guidance අවශ්‍යයි.

HR module එක

HR (Human Resources) යනු ආයතනයේ සේවක තොරතුරු හා වැඩ පාලනයට අදාළය.

දැනගත යුතු කරුණු

- Employee information සහ leave request.
- Roles හා permissions — කවුද මොන data බලන්න පුළුවන්?
- Payroll සහ personal data නිසා privacy ඉතා වැදගත්.



✎ ඔබේ අභ්‍යාසය

Store keeper කෙනෙකුට payroll data වෙනස් කිරීමට අවසර දිය යුතුද?

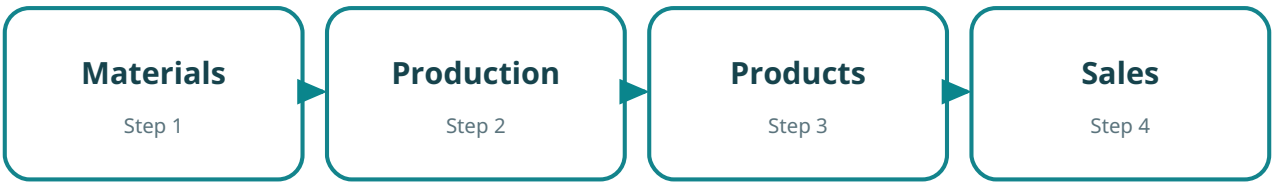
මතක තබාගන්න: මෙම demo එකේ සේවක තොරතුරු ඇතුළත් නොකරයි.

Manufacturing module එක

බිස්කට් හදන factory එකක් වැනි තැනක material → product යන ගමන පාලනය කරයි.

දැනගත යුතු කරුණු

- Raw materials = පිටි, සීනි, පැකට්.
- Production job = මේවායෙන් biscuits සෑදීම.
- Finished goods = විකිණීමට සූදානම් biscuits.



✎ **ඔබේ අභ්‍යාසය**

Cake එකක් හදන raw materials තුනක් නම් කරන්න.

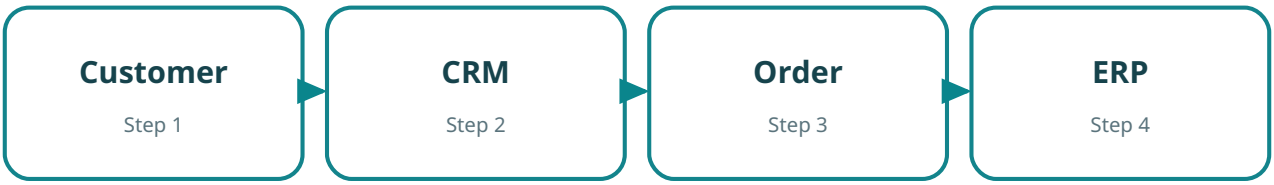
මතක තබාගන්න: MRP කියන්නේ material අවශ්‍යතා ගණනය කිරීමට අදාළ සංකල්පයකි.

CRM එක ERP එකට සම්බන්ධ වන්නේ කොහොමද?

CRM මඟින් customer relations; ERP මඟින් ව්‍යාපාරයේ operations සහ resources පාලනය වේ.

දැනගත යුතු කරුණු

- CRM: customer conversations සහ opportunities.
- ERP: inventory, purchasing, operations, finance.
- Real businesses CRM සහ ERP එකිනෙකට integrate කළ හැක.



✎ ඔබේ අභ්‍යාසය

Customer කෙනෙක් order එකක් කළ විට CRM→ERP යන data එක ගැන උදාහරණයක් දෙන්න.

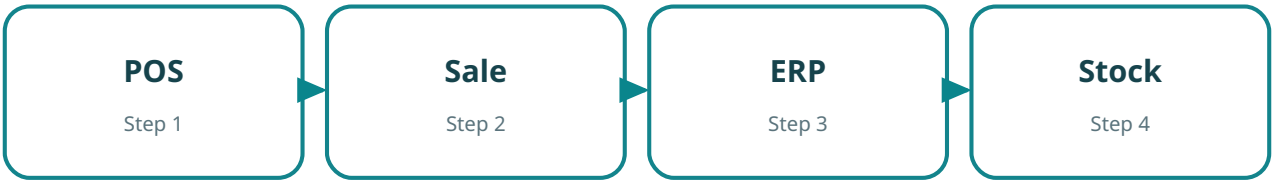
මතක තබාගන්න: අපේ demo එක full CRM system එකක් නොවේ.

POS සහ ERP අතර වෙනස

POS එකක් sale එකක් කරන තැනට වැඩි අවධානයක් දෙනවා. ERP එක ව්‍යාපාරයේ තවත් වැඩ ගොඩක් සම්බන්ධ කරයි.

දැනගත යුතු කරුණු

- POS: items scan, bill, pay සහ receipt.
- ERP: stock, purchases, finance, HR, manufacturing/reporting integration.
- Real company එකක POS විසින් ERP Inventory වෙත sale data යැවිය හැක.



✎ ඔබේ අභ්‍යාසය

POS bill එක save කළ පසු ERP එකේ stock මාරු විය යුතුද?

මතක තබාගන්න: POS එකක් ස්වයංක්‍රීයව සම්පූර්ණ ERP එකක් වෙන්නේ නැහැ.

Cloud ERP සහ Local ERP

Software එක කොහේද run වෙන්නේ කියන එක අනුව deployment model වෙනස් වේ.

දැනගත යුතු කරුණු

- Local/on-premise: ආයතනයේ server එකක ධාවනය විය හැක.
- Cloud ERP: hosted infrastructure හරහා online access.
- 2026දී cloud integration ප්‍රචලිත වුණත් permissions, backups සහ data privacy මූලිකම දේවල්.



✎ ඔබේ අභ්‍යාසය

පාසලේ library එකට internet නැත්නම් cloud ERP එකට ඇති අවදානමක් කියන්න.

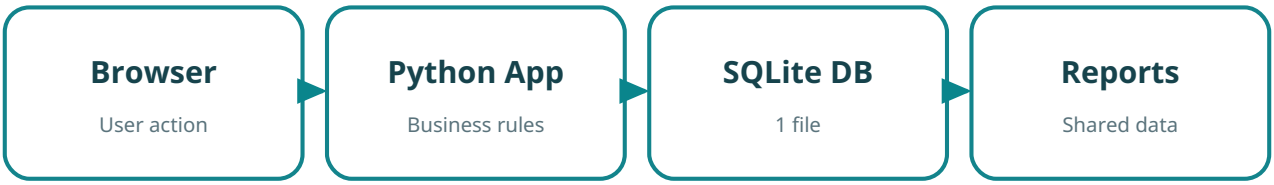
මතක තබාගන්න: අපේ project එක deliberately local-only; cloud-ready බවට පොරොන්දු නොවේ.

එකම database එකක්

Little Star ERP එකේ products, purchases, sales සහ movements එකම SQLite database file එකේ.

දැනගත යුතු කරුණු

- Inventory page එක `products` table එක කියවයි.
- Purchasing page එක `purchases` වලට ලියා `products.stock_qty` වැඩි කරයි.
- Sales page එක `sales` වලට ලියා `products.stock_qty` අඩු කරයි.



✎ **ඔබේ අභ්‍යාසය**

“One database” idea එක arrows තුනකින් ඇඳන්න.

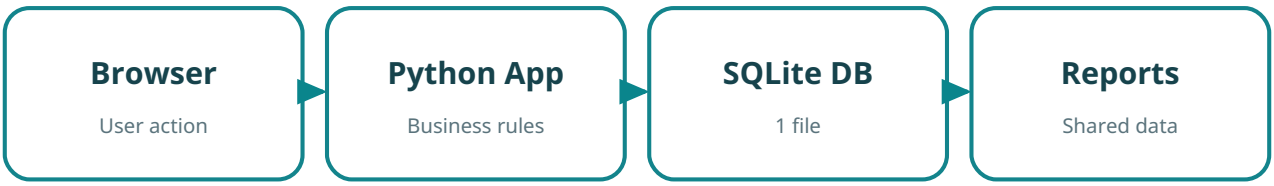
මතක තබාගන්න: Single database concept එක දැනගැනීමෙන් ERP integration ඉතා පහසු වෙයි.

Single source of truth

මෙම වචනය 2026 ව්‍යාපාරික software වලත් වැදගත්.

දැනගත යුතු කරුණු

- එක SKU එකේ නිවැරදි නම හා price එක products table එකේම තබන්න.
- වෙන modules එකම product ID එක භාවිතා කර reference කරයි.
- Duplicate product names නිසා තොරතුරු වෙනස් වීමට ඇති අවදානම අඩු කරයි.



✎ ඔබේ අභ්‍යාසය

“PEN-001” එකම product ID එකෙන් Sales සහ Purchases සම්බන්ධ වන හැටි ලියන්න.

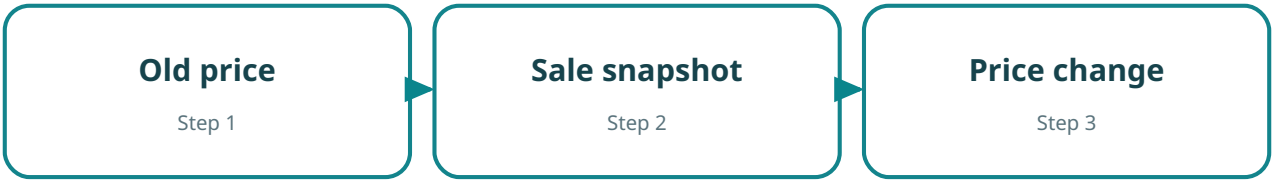
මතක තබාගන්න: එක database එකක ඇති records සියල්ල නිවැරදි කියන සහතිකයක් එයින් නොලැබේ.

සමහර data එක වෙනස් නොකළ යුතුයි

Historical transaction එකක් අද selling price වෙනස් කළා කියලා අලුත් value එකට මාරු නොවිය යුතුයි.

දැනගත යුතු කරුණු

- Sales table එකේ transaction-time unit price එක save කරයි.
- Products table එකේ current selling price එක future sales සඳහායි.
- ඒ නිසා පැරණි විකුණුමක මුදල නිවැරදිව මතක් කළ හැක.



✎ ඔබේ අභ්‍යාසය

අද Rs 80; හෙට Rs 90 නම් ඊයේ receipt එක Rs 90 කරන්න පුළුවන්ද?

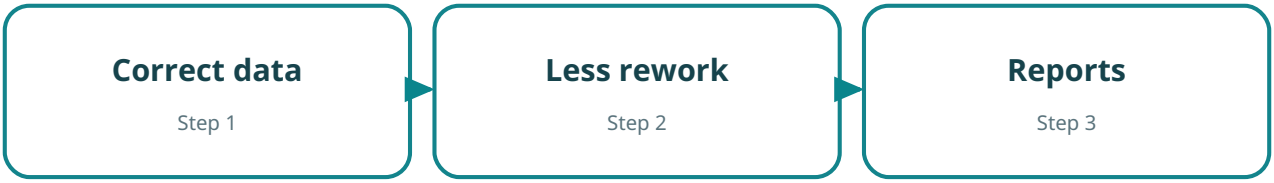
මතක තබාගන්න: Historical facts වෙනස් නොකරන්න; correction workflow එකක් real system වල අවශ්‍යයි.

ERP ප්‍රතිලාභ සහ අභියෝග

ERP එකක් magic software එකක් නොවේ. හොඳ සැලසුමක්, data සහ කණ්ඩායම් පුහුණුව වැදගත්.

දැනගත යුතු කරුණු

- ප්‍රතිලාභ: වැඩ නැවත නැවත නොකිරීම, එක් වාර්තාවක්, වැරදි අඩු වීම.
- අභියෝග: දත්ත පිරිසිදු කිරීම, පුහුණුව, permissions, integration costs.
- කුඩා project එකකින් පටන්ගෙන අවශ්‍ය modules පමණක් වැඩි කිරීම හොඳ ක්‍රමයක්.



✎ **ඔබේ අභ්‍යාසය**

ERP software එක install කළා කියලා සියලු data නිවැරදි වන්නේ ඇයි නැත්තේ?

මතක තබාගන්න: ERP project success මැනෙන්නේ business goal එක ඉටුවූද කියලා.

2026 ERP සහ AI ගැන සරලව

AI assistants සහ analytics ERP වලට එකතු විය හැක. හැබැයි data quality සහ human review අත්‍යවශ්‍යයි.

දැනගත යුතු කරුණු

- AI-assisted search: “දැනට stock අඩු මොනවාද?” කියලා අහන්න.
- Forecasting: පසුගිය sales අනුව අවශ්‍ය stock අනුමාන කිරීම.
- Approval: AI එකක් order එකක් automatically approve කරනවාට කලින් human checks වැදගත්.



> ✎ **ඔබේ අභ්‍යාසය**
> AI forecasting වැරදුණොත් shop එකට ඇතිවිය හැකි ගැටලුවක් ලියන්න.

මතක තබාගන්න: අපේ mini ERP එක AI භාවිතා නොකරනවා; මුලින් reliable data flow ඉගෙනගනිමු.

Unit 01 අභ්‍යාසය

වෙළඳසැලේ Sales, Purchasing, Inventory, Reports modules හතර හඳුනාගන්න පුළුවන්ද?

දැනගත යුතු කරුණු

- ප්‍රශ්නය 1: Supplier කෙනෙකුගෙන් stock ලැබුණා. කුමන module?
- ප්‍රශ්නය 2: 3ක් විකුණා stock අඩු කළා. කුමන module?
- ප්‍රශ්නය 3: මේ මාසයේ revenue පෙන්වන්න. කුමන module?



✎ ඔබේ අභ්‍යාසය

Module flow එක notebook එකක arrows සහ box හතරකින් අඳින්න.

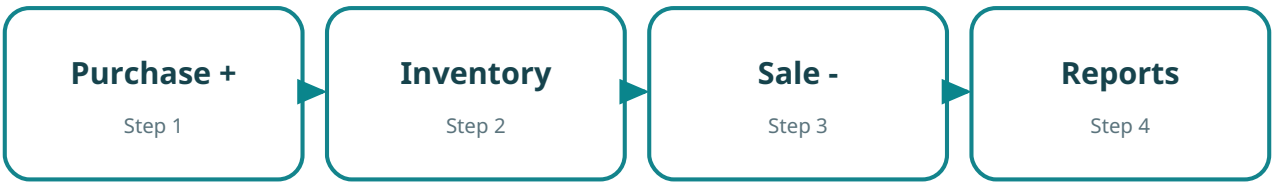
මතක තබාගන්න: පිළිතුරු: Purchasing, Sales, Reports.

Shop එකේ දවසක්

දවසක් අවසානයේ ගණන් නිවැරදිව තිබීමට steps කිහිපයක් සම්බන්ධ වෙන්න ඕන.

දැනගත යුතු කරුණු

- 1. භාණ්ඩ ලැබේ → Purchasing.
- 2. Stock ගණන වෙනස් වේ → Inventory.
- 3. විකුණයි → Sales. 4. තොරතුරු බලයි → Reports.



✎ ඔබේ අභ්‍යාසය

එක pen එකක් purchase කර sale කරන story එක පියවර හතරකින් ලියන්න.

මතක තබාගන්න: එක step එකක් skip කළොත් report එක වෙනස් විය හැක.

Paper receipt vs ERP

කඩදාසි receipt එකක් සුරැකිය හැක, නමුත් stock report එක update කිරීමට අතින් වැඩ අවශ්‍ය විය හැක.

දැනගත යුතු කරුණු

- Paper: sale එක ලියයි; store keeper වෙනම ගණන අඩු කරයි.
- ERP: save sale → purchase/sale table + stock table consistent.
- ඕනෑම technology එකකට data entry checking අවශ්‍යයි.

WITHOUT ERP

Sales notebook: 8 pens

Store notebook: 10 pens

Which one is correct?

WITH ERP

Shared products table: 8

All modules read the same stock

Changes recorded together

✎ ඔබේ අභ්‍යාසය

Sale එක අමතකව stock එක update නොකළොත් පෙන්වෙන wrong number එක සොයන්න.

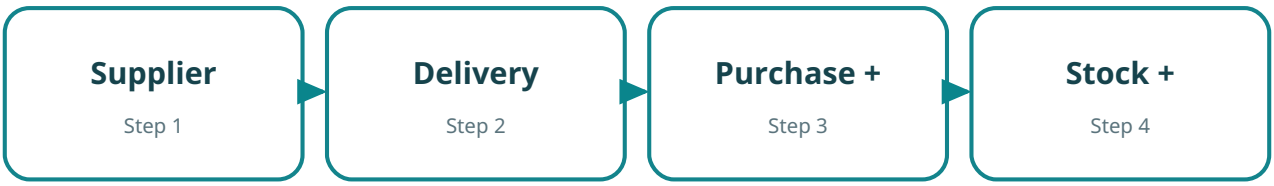
මතක තබාගන්න: ERP data accuracy සඳහා හොඳ processes තවම අවශ්‍යයි.

Stock arithmetic 1: මිලදී ගැනීම

Stock 0කින් පටන් ගෙන supplier delivery එකක් record කරමු.

දැනගත යුතු කරුණු

- Start = 0 pens.
- Receive purchase = +10 pens.
- New stock = 0 + 10 = 10 pens.



✎ ඔබේ අභ්‍යාසය

Start=4, received=7 නම් New stock කීයද?

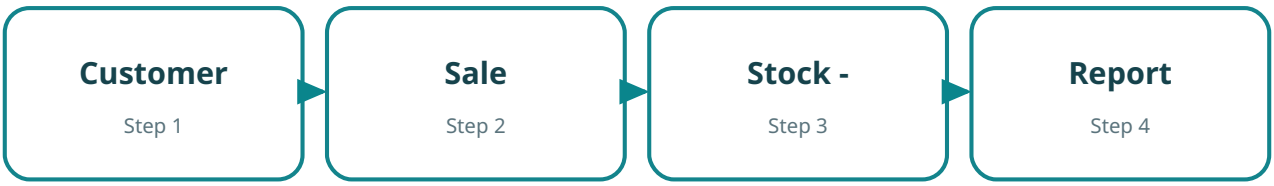
මතක තබාගන්න: Purchase අනුමත කිරීම සහ physically received stock වෙනස් workflows විය හැක.

Stock arithmetic 2: විකිණීම

ඊළඟට pens දෙකක් විකුණමු.

දැනගත යුතු කරුණු

- Stock before sale = 10.
- Sale quantity = 2.
- Stock after sale = 10 - 2 = 8.



✎ ඔබේ අභ්‍යාසය

Stock=8; sale=3 නම් stock කීයද?

මතක තබාගන්න: Negative stock නොවීමට request එක validate කරන්න.

Stock 2යි, customer 5ක් ඉල්ලනවා

මෙය හැම ERP demo එකකම වැදගත් boundary test එකක්.

දැනගත යුතු කරුණු

- Available stock = 2, requested = 5.
- System එක sale එක save නොකර “Not enough stock” කියයි.
- Sales record සහ stock change දෙකම නොවිය යුතුයි.



✎ ඔබේ අභ්‍යාසය

Sale reject කළත් audit trail එකේ වැරදි stock change එකක් ඇතුළත් විය යුතුද?

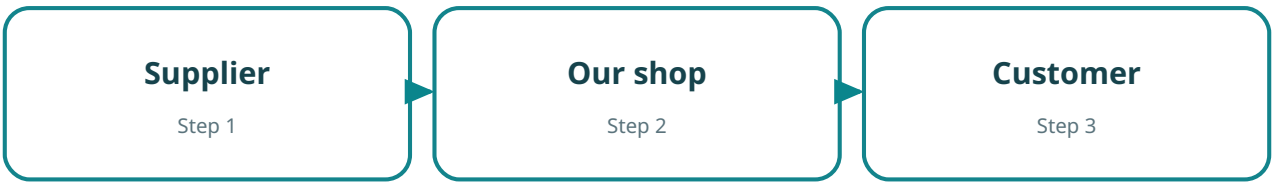
මතක තබාගන්න: Real ERP එකක backorder feature තිබිය හැක; අපේ demo එක එය නොකරයි.

Supplier සහ Customer

Goods සහ data දෙකම කවුරුන් අතර යනවාද කියලා බලමු.

දැනගත යුතු කරුණු

- Supplier → shop වෙත භාණ්ඩ ලබාදේ.
- Shop → customer වෙත භාණ්ඩ විකුණයි.
- Purchasing records supplier; sales records customer name (demo value පමණි).



✎ ඔබේ අභ්‍යාසය

Supplier එකක් සහ customer එකක් අතර වෙනස නිදසුනකින් කියන්න.

මතක තබාගන්න: Real personal data වෙනුවට fictional names පමණක් භාවිතා කරන්න.

SKU නැත්නම් මොකද වෙන්නේ?

“Blue Pen” සහ “Blue pen” එකම භාණ්ඩයද කියලා අලුත් දෙකක් වීම වැළැක්වීමට unique SKU උදව් කරයි.

දැනගත යුතු කරුණු

- SKU යනු product identification code එකකි.
- DB `UNIQUE` rule එක duplicate SKU එක block කරයි.
- Product ID වෙන tables වල reference කරන්න පහසුයි.



> ✎ **ඔබේ අභ්‍යාසය**
>
> Ruler එකට RULER-001 යන SKU එක යෝජනා කරන්න.

මතක තබාගන්න: SKU code එක ව්‍යාපාරයේ naming policy එකකට ගැළපිය යුතුයි.

Stock Movement Log

Stock ගණන වෙනස් වූ විට එය ඇයි වෙනස් වූයේ කියලා දැනගන්න history එක වැදගත්.

දැනගත යුතු කරුණු

- Purchase 10 → movement +10, reason PURCHASE.
- Sale 2 → movement -2, reason SALE.
- Stock = සියලු movement changes වල එකතුව (අපේ zero-start demo එකේ).



✎ ඔබේ අභ්‍යාසය

Movement table එකට [+10, -2, -3] ලියා total කීයද හොයන්න.

මතක තබාගන්න: Audit trail එකක් stock discrepancies සොයන්න උදව් කරයි.

Low-stock alert

භාණ්ඩ ඉවර වෙන්න කලින් warning එකක් තිබුණොත් නැවත order එකක් ගැන සලකා බලන්න පුළුවන්.

දැනගත යුතු කරුණු

- Product reorder level = 5.
- Actual stock = 4 නම් low-stock warning.
- Our demo condition: stock_qty <= reorder_level.



✎ ඔබේ අභ්‍යාසය

Level=5 නම් stock 5ක් තිබුණාම warning පෙන්වෙනවාද?

මතක තබාගන්න: Alert එකක් automate කළත් අලුත් purchase approval මිනිසා පරීක්ෂා කළ යුතුයි.

Price vs Cost vs Revenue

ව්‍යාපාරයේ රුපියල් ගණන් තුනක් එකක් ලෙස නොසලකන්න.

දැනගත යුතු කරුණු

- Selling price: customer ගෙවන unit price.
- Purchase cost: supplier ට ගෙවන unit cost.
- Sales revenue: විකුණූ unit_price × quantity; profit නොවේ.



✎ ඔබේ අභ්‍යාසය

Pen එකක් Rs 50කට ගෙන Rs 80කට විකිණුවොත් revenue සහ purchase cost වෙනම ලියන්න.

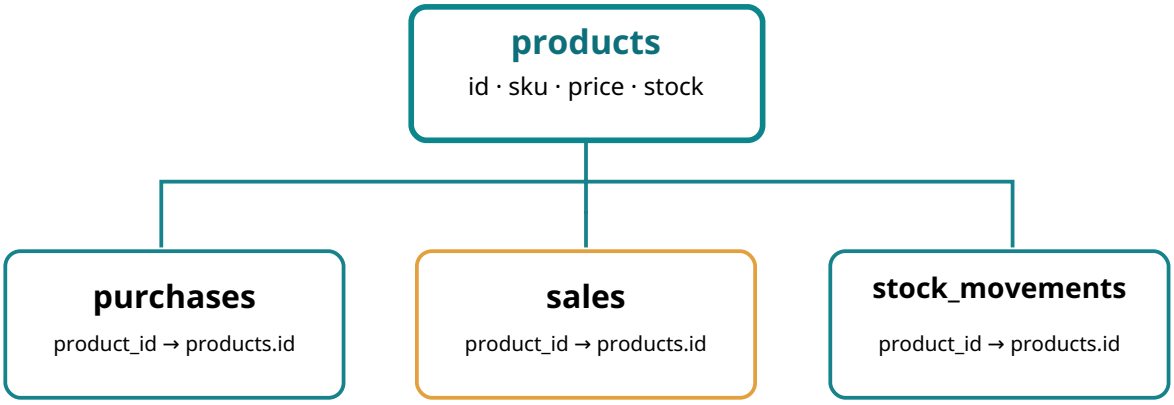
මතක තබාගන්න: Real profit ගණනය කිරීමට විකුණූ භාණ්ඩයේ actual cost සහ වෙනත් වියදම් අවශ්‍යයි.

Database tables හතර

එක් SQLite file එකක tables හතරක්. එක එක table එක නිශ්චිත data එකක් රඳවා තබයි.

දැනගත යුතු කරුණු

- `products`: නිෂ්පාදන නාම, SKU, current stock.
- `purchases`, `sales`: හුවමාරු records.
- `stock_movements`: stock +/− history.



✎ ඔබේ අභ්‍යාසය

Table name හතරක් සහ ඒවායේ fields දෙක බැගින් ලියන්න.

මතක තබාගන්න: Product ID එක tables අතර සම්බන්ධතා සාදයි.

Primary Key සහ Foreign Key

Database tables දෙකක් සම්බන්ධ කිරීමේ මූලිකම අදහස මේකයි.

දැනගත යුතු කරුණු

- Primary Key `products.id`: product එකකට එක unique number.
- Foreign Key `sales.product_id`: අදාළ product එකේ ID.
- මෙයින් නොපවතින product එකකට sale එකක් දාන්න වළක්වන්න පුළුවන්.



✎ ඔබේ අභ්‍යාසය

products.id=1 නම් sales.product_id=1 යන්නෙන් අර්ථය කුමක්ද?

මතක තබාගන්න: Database foreign key check එක enable කර තිබිය යුතුයි.

Transaction කියන්නේ මොකක්ද?

ERP එකක වැඩ දෙකක් එකටම සාර්ථක වීම හෝ එකටම අවලංගු වීම වැදගත්.

දැනගත යුතු කරුණු

- Sale record insert කර stock deduct නොවුණොත් data mismatch.
- Transaction එකක් දෙකම complete කළොත් COMMIT.
- මැද error නම් ROLLBACK කර මුල් තත්ත්වයට ආපසු යන්න.



✎ ඔබේ අභ්‍යාසය

Sale save වුණා, stock අඩු වුණේ නැහැ — ඇයි මෙය භයානකද?

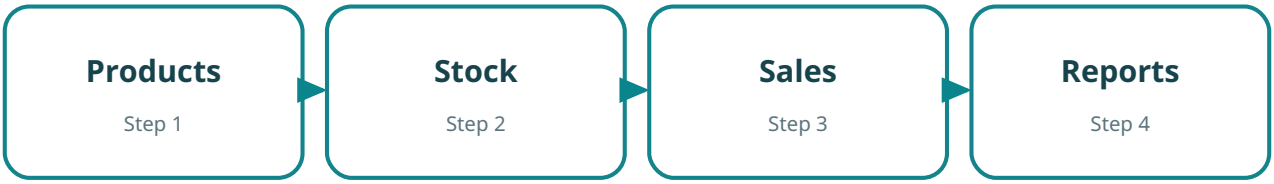
මතක තබාගන්න: Transactions සහ validation inventory correctness සඳහා ප්‍රධාන.

Reports එක shared data පෙන්වනවා

Report එකකට data අලුතින් type කරන්න ඕන නැහැ. Tables එකතුවෙන් සාරාංශය ලැබෙනවා.

දැනගත යුතු කරුණු

- Products table: stock සහ low-stock count.
- Sales table: sale count සහ revenue.
- Stock movements: ඇත්තටම stock වෙනස් වූ හේතු.



✎ ඔබේ අභ්‍යාසය

Sales transaction එක Reports වෙත ගමන් කරන diagram එකක් අඳින්න.

මතක තබාගන්න: කුමන data ගණන් ගන්නවාද කියලා report label එක පැහැදිලිව ලියන්න.

Practice: mini stock book

Code නොමැතිව notebook එකේ mini ERP simulation එකක් කරමු.

දැනගත යුතු කරුණු

- Start pen=0; purchase +10; sale -2; purchase +3.
- Final stock = 11.
- ඊළඟට notebooks 0 සිට +5, -1 කියලා වෙන් කර ගණනය කරන්න.



✎ ඔබේ අභ්‍යාසය

ඔබේ ගණන් two-column ledger එකක් ලෙස ලියන්න.

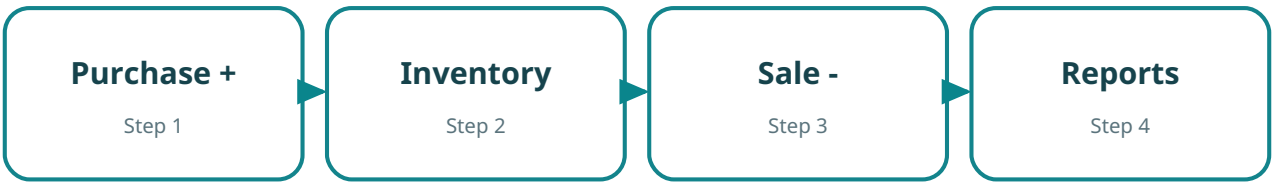
මතක තබාගන්න: Module එකක change එක අනෙක් modules වල data වලට බලපානවා.

Unit 02 quick check

ERP එක හොඳින් තේරුම් ගන්න math සහ workflow දෙකම භාවිතා කරන්න.

දැනගත යුතු කරුණු

- 1. Stock 6 තිබුණාම 9 විකුණන ලෙස request කළොත් කුමක් වෙයිද?
- 2. Purchase +4, Sale -3, Purchase +2 → net change කීයද?
- 3. Sales revenue = profit කියලා ලියන්න පුළුවන්ද?



✎ ඔබේ අභ්‍යාසය

පිළිතුරු: reject sale; net +3; බැහැ.

මතක තබාගන්න: නැවත අසන්න: එකම stock value එක all pages වල පෙනෙන්නේ ඇයි?

Software එකක් හදන ගමන

ERP එක හදනකොට SDLC — Software Development Life Cycle — මඟින් project එක පිළිවෙලට පාලනය කරමු.

දැනගත යුතු කරුණු

- Discover → Requirements → Design → Build → Test → Release → Maintain.
- අද අපේ sample shop එකේ ගැටලු දැනගනිමු.
- ලියපු තොරතුරු documents ලෙස තබා ගැනීමෙන් tester ටත් වැඩ පහසුයි.



✎ ඔබේ අභ්‍යාසය

SDLC steps 7 වරක් ඔබේ notebook එකේ අඳින්න.

මතක තබාගන්න: Documentation code එකට පසුව දමන දෙයක් පමණක් නොවේ.

Problem Statement

“Shop owner ට stock සහ sales එකට පෙන්වන සරල screen එකක් නැහැ” කියලා problem එක ලියමු.

දැනගත යුතු කරුණු

- Existing problem: separate notebooks; stale stock counts.
- Impact: overselling සහ වැරදි reporting.
- Goal: one shared database and simple forms.

WITHOUT ERP	WITH ERP
Sales notebook: 8 pens	Shared products table: 8
Store notebook: 10 pens	All modules read the same stock
Which one is correct?	Changes recorded together

✎ ඔබේ අභ්‍යාසය

කඩදාසි හෝ Excel භාවිතයෙන් ඇතිවන එක් pain point එකක් ලියන්න.

මතක තබාගන්න: User problem එක පළමුව තේරුම් ගන්න.

Stakeholders හඳුනාගනිමු

Project එකෙන් බලපෑමක් ඇති අය stakeholders.

දැනගත යුතු කරුණු

- Shop owner: overview; store keeper: purchase/stock.
- Cashier: sales; teacher/student: learning workflow.
- Tester: correct validation සහ stock consistency.



✎ ඔබේ අභ්‍යාසය

කෙනා, අවශ්‍යතාව, module යන columns තුනක table එකක් හදාගන්න.

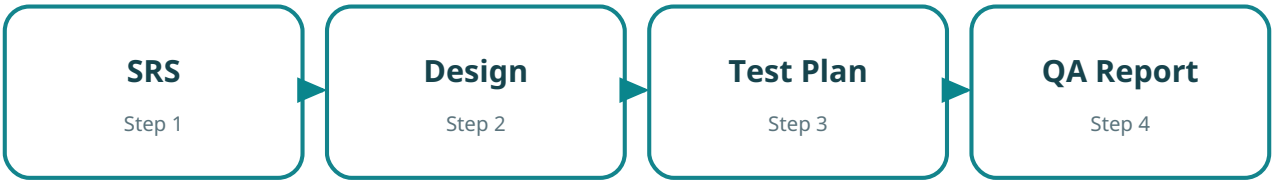
මතක තබාගන්න: Real business roles වෙනසක් ඇති නිසා permissions ද අවශ්‍යයි; demo එකේ නැහැ.

SRS කියන්නේ මොකක්ද?

SRS = Software Requirements Specification. කරන්න යන්නේ මොකක්ද කියලා පැහැදිලි ලිඛිත එකඟතාවක්.

දැනගත යුතු කරුණු

- Functional requirements = app කරන වැඩ.
- Non-functional requirements = වේගය, ආරක්ෂාව, usability වැනි ගුණාංග.
- Out of scope = මේ version එකේ නොකරන වැඩ.



✎ ඔබේ අභ්‍යාසය

SRS එකේ FR-01 = “View products” ලෙස ලියන්න.

මතක තබාගන්න: Project ZIP එකේ `docs/SRS.md` ඇත.

Functional Requirements

Little Star ERP සඳහා ප්‍රධාන FR list එක සකස් කරමු.

දැනගත යුතු කරුණු

- FR-01 View products/stock; FR-02 Add product.
- FR-03 Record purchase and increase stock; FR-04 Record sale and decrease stock.
- FR-05 Show reports; FR-06 Record stock movements.



✎ ඔබේ අභ්‍යාසය

“Reject overselling” සඳහා FR-07 ලෙස requirement එකක් ලියන්න.

මතක තබාගන්න: Requirement එකක් tester කෙනෙකුට verify කළ හැකි ලෙස ලියන්න.

Non-functional Requirements

Feature එකක් කරන දේට අමතරව එය කොහොම වැඩ කළ යුතුද කියලා බලමු.

දැනගත යුතු කරුණු

- Security: locally bind to 127.0.0.1, parameterized SQL, escape HTML.
- Reliability: stock transactions all-or-nothing.
- Usability: user-friendly messages; simple navigation.



✎ ඔබේ අභ්‍යාසය

“අන්තර්ජාලය නැතිව වැඩ” කියන requirement එක functional ද non-functional ද?

මතක තබාගන්න: Local Python + SQLite example එක අමතර pip packages අවශ්‍ය නැති ලෙස නිර්මාණය කර ඇත.

User Story එකක් ලියමු

User කෙනෙකුට ලැබෙන වටිනාකම පෙන්වමු.

දැනගත යුතු කරුණු

- As a store keeper, I want to receive new stock so that inventory is accurate.
- As a cashier, I want to record a sale so that stock decreases.
- As a shop owner, I want a dashboard so I can monitor low stock.



✎ ඔබේ අභ්‍යාසය

“As a … I want … so that …” pattern එකෙන් report user story එකක් ලියන්න.

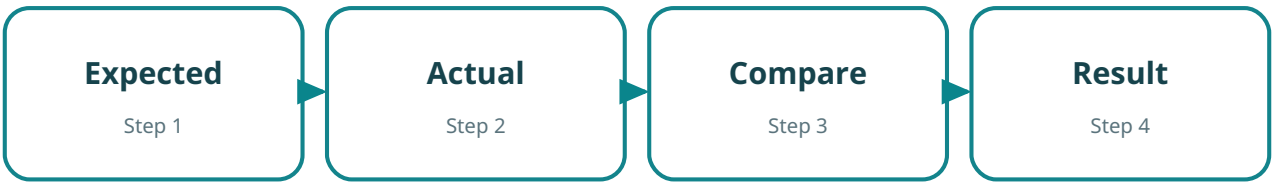
මතක තබාගන්න: User story සහ technical tasks සම්බන්ධ නමුත් එකම දේ නොවේ.

Acceptance Criteria

වැඩ කළා කියලා පිළිගන්න conditions ලියමු.

දැනගත යුතු කරුණු

- Given 0 pens, when receive 10, then stock = 10.
- Given stock 10, when sell 2, then stock = 8.
- Given stock 2, when sell 5, then error + unchanged data.



✎ ඔබේ අභ්‍යාසය

ඔබේ අලුත් FR එකට Given/When/Then එකක් ලියන්න.

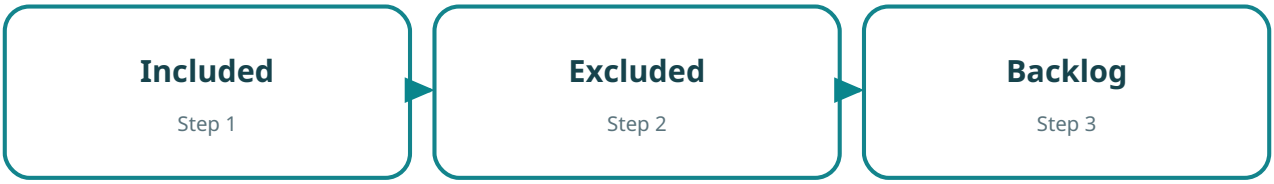
මතක තබාගන්න: Tests වල expected result එක requirements වලින් එනවා.

Scope එක සීමා කරමු

අපේ project එක ඉගෙනීමට. ඒ නිසා everything-in-one enterprise app එකක් කරන්න යන්නේ නැහැ.

දැනගත යුතු කරුණු

- In scope: product master, purchase, sale, stock ledger, reports.
- Out of scope: payment gateway, taxes, payroll, multi-store inventory.
- Out of scope: production security, customer personal data, real accounting.



✎ **ඔබේ අභ්‍යාසය**

“Barcode scanner” feature එක future backlog එකකට ලියන්න.

මතක තබාගන්න: Scope හිස්ව නැතිව, manageable size එකක project එකක් තෝරාගැනීම වටිනවා.

UI Wireframe එකක්

Code ලිවීමට කලින් browser page එක sketch කරන්න.

දැනගත යුතු කරුණු

- Top navigation: Dashboard, Products, Purchasing, Sales, Reports.
- Dashboard: Products, Items in Stock, Low Stock, Revenue.
- Forms: labels, inputs, Save buttons; messages.



✎ ඔබේ අභ්‍යාසය

Dashboard page එක කඩදාසියක box සහ labels වලින් අඳින්න.

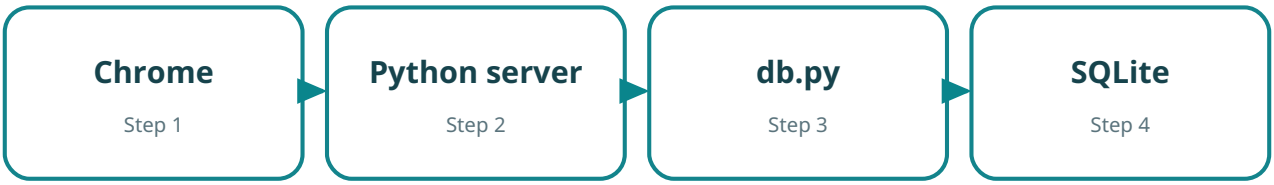
මතක තබාගන්න: අලංකාරයට වඩා clear information order සහ readability වැදගත්.

Architecture Diagram

පරිශීලකයාගේ click එක Python server එකට යයි; server එක database එක update කර page එක නැවත පෙන්වයි.

දැනගත යුතු කරුණු

- Chrome/Edge browser → local Python HTTP server.
- Server → db.py functions → SQLite database.
- Response → HTML page + CSS.



✎ **ඔබේ අභ්‍යාසය**

Browser, Server, Database boxes තුන arrows සමඟ අඳින්න.

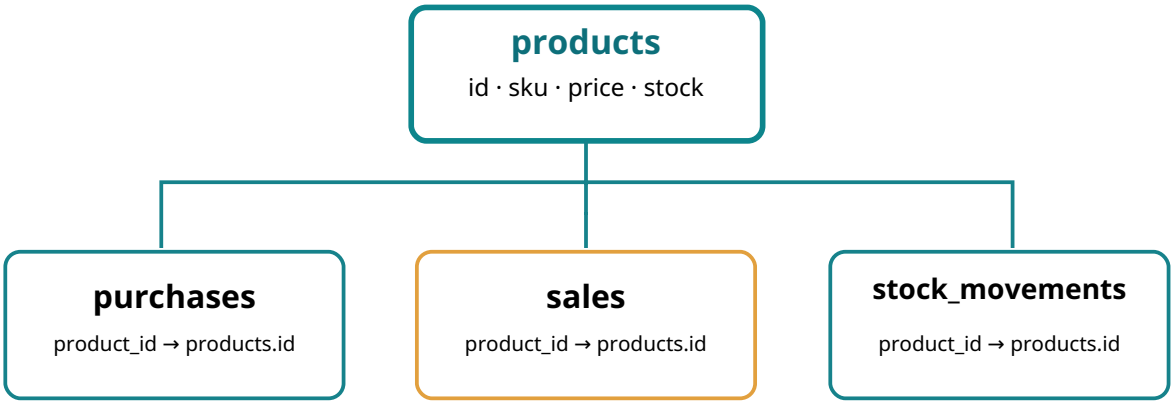
මතක තබාගන්න: Local demo එකට Internet හෝ public hosting අවශ්‍ය නැහැ.

Database ER Diagram

Tables වල relationships model කරගනිමු.

දැනගත යුතු කරුණු

- products.id එක purchases.product_id සමඟ link වේ.
- products.id එක sales.product_id සමඟ link වේ.
- stock_movements එකේ product_id ද products සමඟ link වේ.



✎ ඔබේ අභ්‍යාසය

Product එකක් විකුණු අවස්ථා 10ක් තිබිය හැක; One-to-Many arrow එක අඳින්න.

මතක තබාගන්න: Diagram එක database constraints තේරුම් ගන්නත් උදව් කරයි.

Risk සහ QA Plan

Developer code ලියන වෙලාවටම potential bugs ගැන සිතනවා.

දැනගත යුතු කරුණු

- Risk: oversell → validation + transaction.
- Risk: duplicate SKU → unique constraint + friendly message.
- Risk: invalid money → Decimal conversion to integer cents.



✎ ඔබේ අභ්‍යාසය

“Stock 0ක් වෙද්දී Sale button click” සඳහා expected result එක ලියන්න.

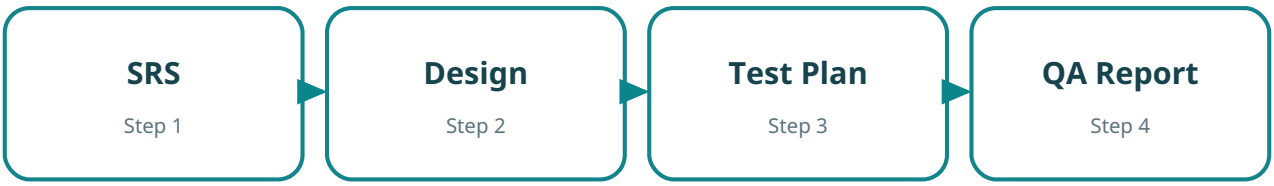
මතක තබාගන්න: QA යනු testing පමණක් නොව quality process එක සම්පූර්ණයෙන් සලකා බැලීමයි.

Unit 03 Review

SRS සහ diagrams හදපු පසු coding එක වඩා පහසුයි.

දැනගත යුතු කරුණු

- Problem statement එක යනු project එක අවශ්‍ය ඇයි යන්නයි.
- SRS එක කරන වැඩ සහ non-functional quality සඳහන් කරයි.
- Testing plan එක requirements පරීක්ෂා කරන steps දක්වයි.



✎ ඔබේ අභ්‍යාසය

FR-04 සහ TC-05 එකිනෙක සම්බන්ධව ලියන්න.

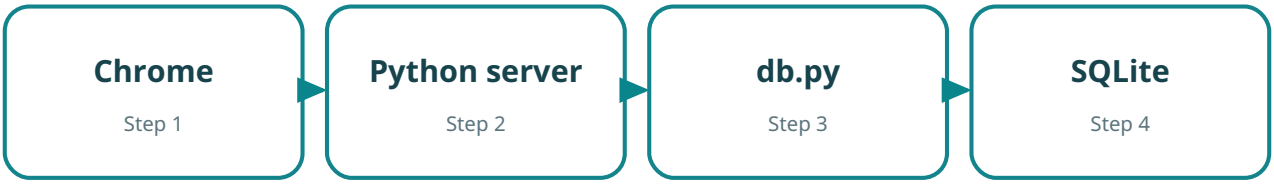
මතක තබාගන්න: Source ZIP එකේ SRS, QA Plan, Data Dictionary සහ SDLC Roadmap documents ඇත.

Python භාවිතා කරන්නේ ඇයි?

Python beginner-friendly language එකක්; එයට built-in SQLite සහ HTTP server features ඇත.

දැනගත යුතු කරුණු

- Simple readable syntax ඉගෙනගන්න පහසුයි.
- `sqlite3` මඟින් database file එකක් manage කළ හැක.
- `http.server` මඟින් local learning web app එකක් serve කළ හැක.



✎ ඔබේ අභ්‍යාසය

Python එක browser app එකේ කොටස් තුනට කෙසේ උදව් කරනවාද කියන්න.

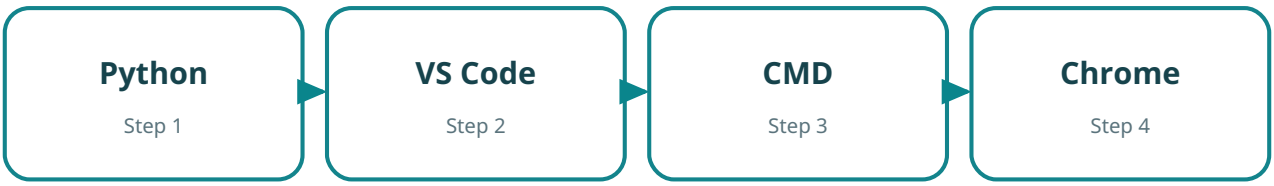
මතක තබාගන්න: Built-in HTTP server එක production වෙබ් hosting සඳහා නිර්දේශ නොකරයි.

අවශ්‍ය software

අපේ project එක Windows 10/11, Python සහ browser සමඟ local practice කළ හැක.

දැනගත යුතු කරුණු

- Python 3.10+; 3.12 හෝ නවතම supported version යෝග්‍යයි.
- Visual Studio Code වැනි text/code editor එකක්.
- Chrome හෝ Edge browser + Windows Command Prompt.



✎ ඔබේ අභ්‍යාසය

Python version එක `py --version` මඟින් පරීක්ෂා කරන්න.

මතක තබාගන්න: 2026දී install කරන version එක python.org හි supported release එකක් බව බලන්න.

Python install පරීක්ෂාව

CMD තුළ commands type කරන විට copy-paste කළත් අර්ථය තේරුම් ගන්න.

දැනගත යුතු කරුණු

- `py --version` = Python launcher version.
- `python --version` = PATH හි Python version.
- `py` command නැත්නම් Python install / PATH බලන්න.

```
Windows Command Prompt                                    _ □ ×

C:\Users\Student> py --version
C:\Users\Student> python --version
```

> ✎ **ඔබේ අභ්‍යාසය**
>
> CMD එක open කර version output එක notebook එකේ ලියන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Project ZIP extract කරන්න

Source code ZIP එක unpack කර project root folder එකට යන්න.

දැනගත යුතු කරුණු

- Folder: `EGOTECHWORLD_Little_Star_ERP_2026`.
- මේ folder එකේ `app.py`, `db.py`, `schema.sql` ඇත.
- `data/little_star.sqlite3` file එක පළමු run එකේදී නිර්මාණය වේ.

VS Code · EXPLORER

```
EGOTECHWORLD_Little_Star_ERP_2026/
  |-- app.py
  |-- db.py
  |-- schema.sql
  |-- tests.py
  |-- run_windows.bat
  |-- assets/
  |    `-- style.css
  |-- docs/
  |    |-- SRS.md
  |    |-- QA_Test_Plan.md
  |    `-- Data_Dictionary.md
  `-- data/  (created when app runs)
```

✎ ඔබේ අභ්‍යාසය

Source ZIP extract කර VS Code → Open Folder කරන්න.

මතක තබාගන්න: ZIP එක ඇතුළේ ලිපිගොනු edit නොකර කලින් extract කරන්න.

VS Code Explorer එක

Code files එකිනෙක සම්බන්ධ වෙන ආකාරය folder tree එකෙන් බලන්න.

දැනගත යුතු කරුණු

- `app.py` = web pages + form actions.
- `db.py` = database functions.
- `schema.sql` = table structure. `tests.py` = automated tests.

VS Code · EXPLORER

```
EGOTECHWORLD_Little_Star_ERP_2026/
  |-- app.py
  |-- db.py
  |-- schema.sql
  |-- tests.py
  |-- run_windows.bat
  |-- assets/
  |    `-- style.css
  |-- docs/
  |    |-- SRS.md
  |    |-- QA_Test_Plan.md
  |    `-- Data_Dictionary.md
  `-- data/  (created when app runs)
```

✎ ඔබේ අභ්‍යාසය

Explorer එකෙන් app.py සහ db.py වෙන වෙනම open කරන්න.

මතක තබාගන්න: Project එකේ files අඩු වුණත් responsibilities වෙන් කර ඇත.

CMD එකෙන් Project run කරන්න

Source folder එක ඇතුළේ CMD open කර command එකක් execute කරමු.

දැනගත යුතු කරුණු

- `cd` = project directory එකට යන්න.
- `py app.py` = local Python server start කරන්න.
- `Ctrl+C` = server එක නවත්වන්න.

```
Windows Command Prompt                                    _ □ ×

C:\Users\Student> cd Desktop\EGOTECHWORLD_Little_Star_ERP_2026
C:\Users\Student> py app.py
```

> ✎ **ඔබේ අභ්‍යාසය**
>
> ඔබේ folder path එකට ගැලපෙන `cd` command ලියන්න.

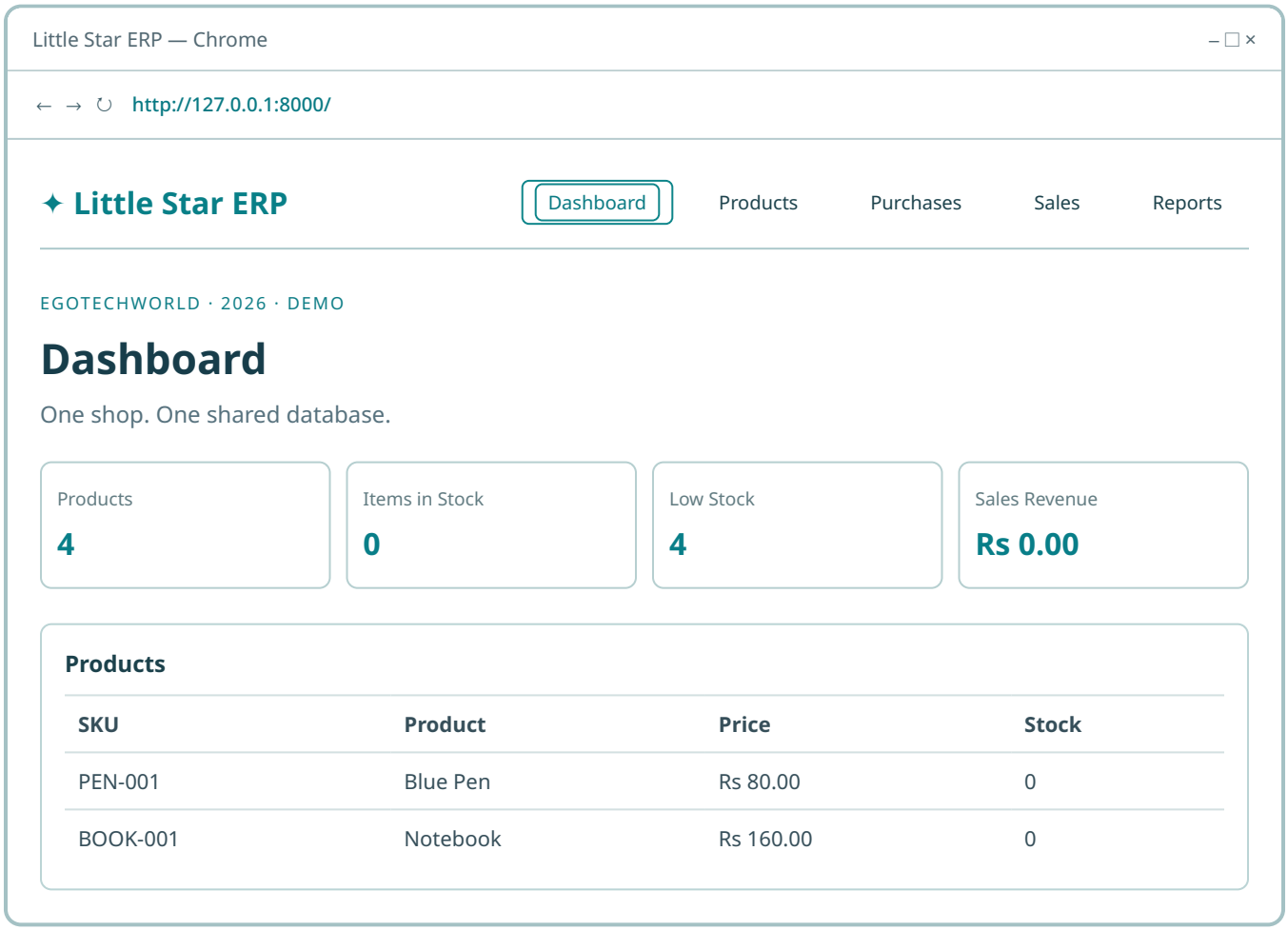
මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Chrome එකෙන් ERP open කරමු

Server start වුණාට පසු browser address bar එකේ local URL type කරන්න.

දැනගත යුතු කරුණු

- URL: `http://127.0.0.1:8000`.
- මේ address එක ඔබේ computer එකම පෙන්වනවා.
- Dashboard page එකේ Products, Stock, Low Stock සහ Revenue numbers පෙනේ.



Illustrative Chrome-style preview using the sample app's fields; live screen may vary.

✎ ඔබේ අභ්‍යාසය

Page එක open කර navigation labels පහ කියවන්න.

මතක තබාගන්න: `localhost` links ඔබේ computer එකේ පමණක් ක්‍රියා කරනවා.

Python print සහ variables

ERP project එකට කලින් Python ගණන් සරල code එකකින් ඉගෙනගනිමු.

දැනගත යුතු කරුණු

- `print()` කියන්නේ screen එකට message යැවීම.
- Variable එකක් අගයක් මතක තබාගන්නා නමක්.
- `quantity = 10` යනු භාණ්ඩ 10ක් නම් කිරීම.



> ✎ **ඔබේ අභ්‍යාසය**
>
> Quantity 10 වෙනුවට 15 කර output එක වෙනස් වන හැටි බලන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Data Types: text සහ number

ERP එකේ Product name text; stock whole number; price money value එකක්.

දැනගත යුතු කරුණු

- `str`: "Blue Pen", "PEN-001".
- `int`: 0, 5, 10 වැනි whole numbers.
- Money සඳහා demo එක integer cents භාවිතා කර binary floating-point rounding අවම කරයි.



✎ ඔබේ අභ්‍යාසය

"Notebook" සහ 16000 අතර type වෙනස කියන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

If Condition – Stock ඇත්ද?

Code එකකට තීරණ ගැනීමට if statement උපකාරී වේ.

දැනගත යුතු කරුණු

- If available stock >= requested units → sale allowed.
- Else → friendly error එකක්.
- මේ rule එක අපේ `sell_stock()` තුළ තිබෙනවා.



✎ ඔබේ අභ්‍යාසය

Stock=2, requested=5 නම් print වෙන්නේ මොකක්ද?

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Functions කියන්නේ පුංචි වැඩකරුවන්

Same code එක නැවත නැවත නොලියා function එකක් සාදා call කළ හැක.

දැනගත යුතු කරුණු

- `def` මඟින් function එක හඳුන්වයි.
- Input parameters වලට numbers යවන්න.
- `return` result එක ආපසු ලබා දෙයි.



> ✎ **ඔබේ අභ්‍යාසය**
>
> Number දෙකක් add කරන function එකක් ඔබම type කරන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

List සහ Loop

බහු products එකින් එක කියවීම සඳහා list සහ for loop ඉතා ප්‍රයෝජනවත්.

දැනගත යුතු කරුණු

- List එකක item කිහිපයක් එකට ගබඩා කළ හැක.
- `for` loop එකෙන් සෑම item එකකටම යන්න පුළුවන්.
- ERP product list එක DB rows වලින් සාදනු ලැබේ.



✎ ඔබේ අභ්‍යාසය

Eraser කියන item එක list එකට එකතු කරන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Dictionary සහ Table Row

SQLite query එකෙන් ලැබෙන row එක key-value mapping එකක් ලෙස භාවිතා කළ හැක.

දැනගත යුතු කරුණු

- `product["name"]` මඟින් name value එක ගනී.
- `product["stock_qty"]` මඟින් stock quantity ගනී.
- HTML table එකට මේ values escape කරලා දක්වන්න.



✎ ඔබේ අභ්‍යාසය

Dictionary එකකට `sku` key එකක් එක් කරන්න.

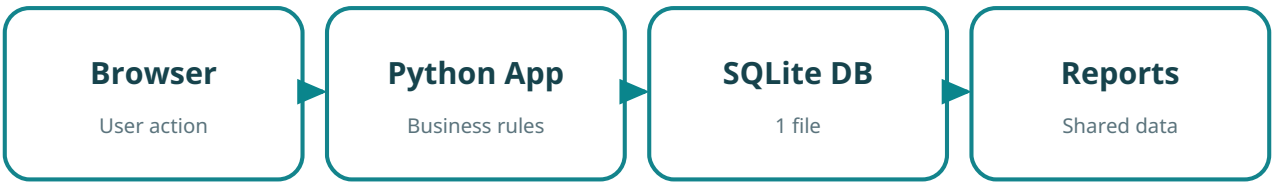
මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

SQLite කියන්නේ මොකක්ද?

SQLite යනු Python සමඟම භාවිතා කළ හැකි lightweight database එකකි.

දැනගත යුතු කරුණු

- මේ demo එකට MySQL server එකක් අවශ්‍ය නැහැ.
- Database file එක project එකේ `data/` තුළ රඳවා තබයි.
- SQL queries මඟින් records add, read, update කරයි.



✎ ඔබේ අභ්‍යාසය

Notebook එකක “Database → Tables → Rows → Columns” ලෙස ලියන්න.

මතක තබාගන්න: Real ERP එකකට database design, backups සහ scale වෙනම සැලසුම් කළ යුතුයි.

SQL SELECT – data කියවමු

SQL = Structured Query Language. Tables වලින් තොරතුරු ඉල්ලීමට SELECT භාවිතා කරයි.

දැනගත යුතු කරුණු

- `SELECT name, stock_qty FROM products` → නම සහ stock.
- `ORDER BY name` → අකාරාදි අනුපිළිවෙල.
- Product page එකට database values එන හැටි බලන්න.



✎ ඔබේ අභ්‍යාසය

Stock පමණක් අවශ්‍ය නම් SELECT එකට කුමන column එක දෙන්නද?

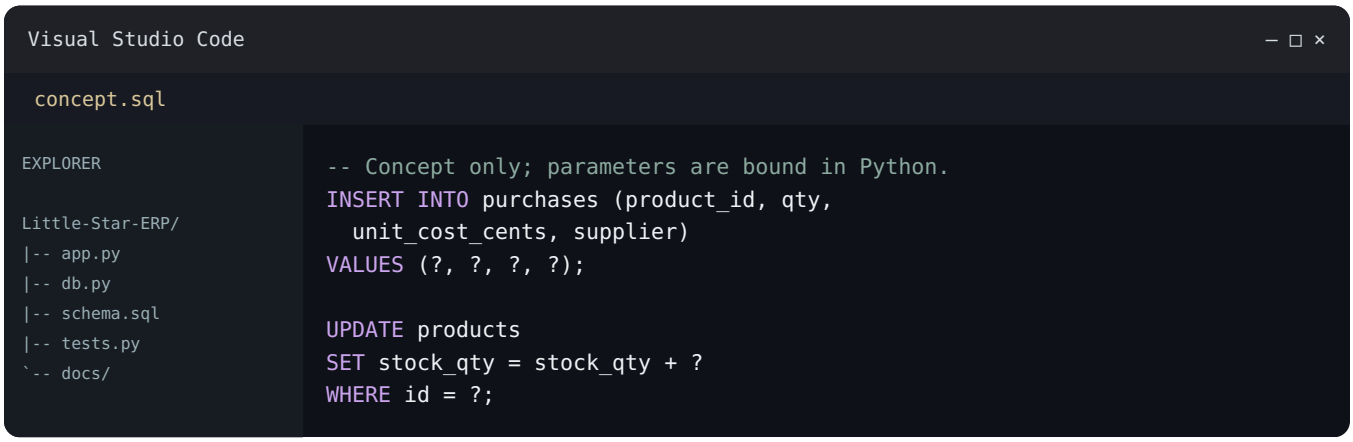
මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

SQL INSERT සහ UPDATE

Purchasing ක්‍රියාවකදී new purchase row එකක් සහ stock update එකක් අවශ්‍යයි.

දැනගත යුතු කරුණු

- INSERT = අලුත් row එකක් එකතු කරන්න.
- UPDATE = දැනට තිබෙන row එකක් වෙනස් කරන්න.
- එකම transaction එකක දෙකම කරන්න.



> ✎ **ඔබේ අභ්‍යාසය**
>
> UPDATE fail වුණොත් INSERT එකත් rollback කළ යුත්තේ ඇයි?

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Local App Security

Learning app එකකටත් basic security rules තිබීම හොඳයි.

දැනගත යුතු කරුණු

- Project එක Internet එකට publish නොකරන්න; 127.0.0.1 පමණක්.
- Database queries `?` placeholders භාවිතා කර user input safe bind කරයි.
- HTML escape සහ local demo CSRF token check ද ඇතුළත්.



✎ ඔබේ අභ්‍යාසය

Public server එකකට login නැති ERP app එකක් upload කිරීමේ අවදානම කියන්න.

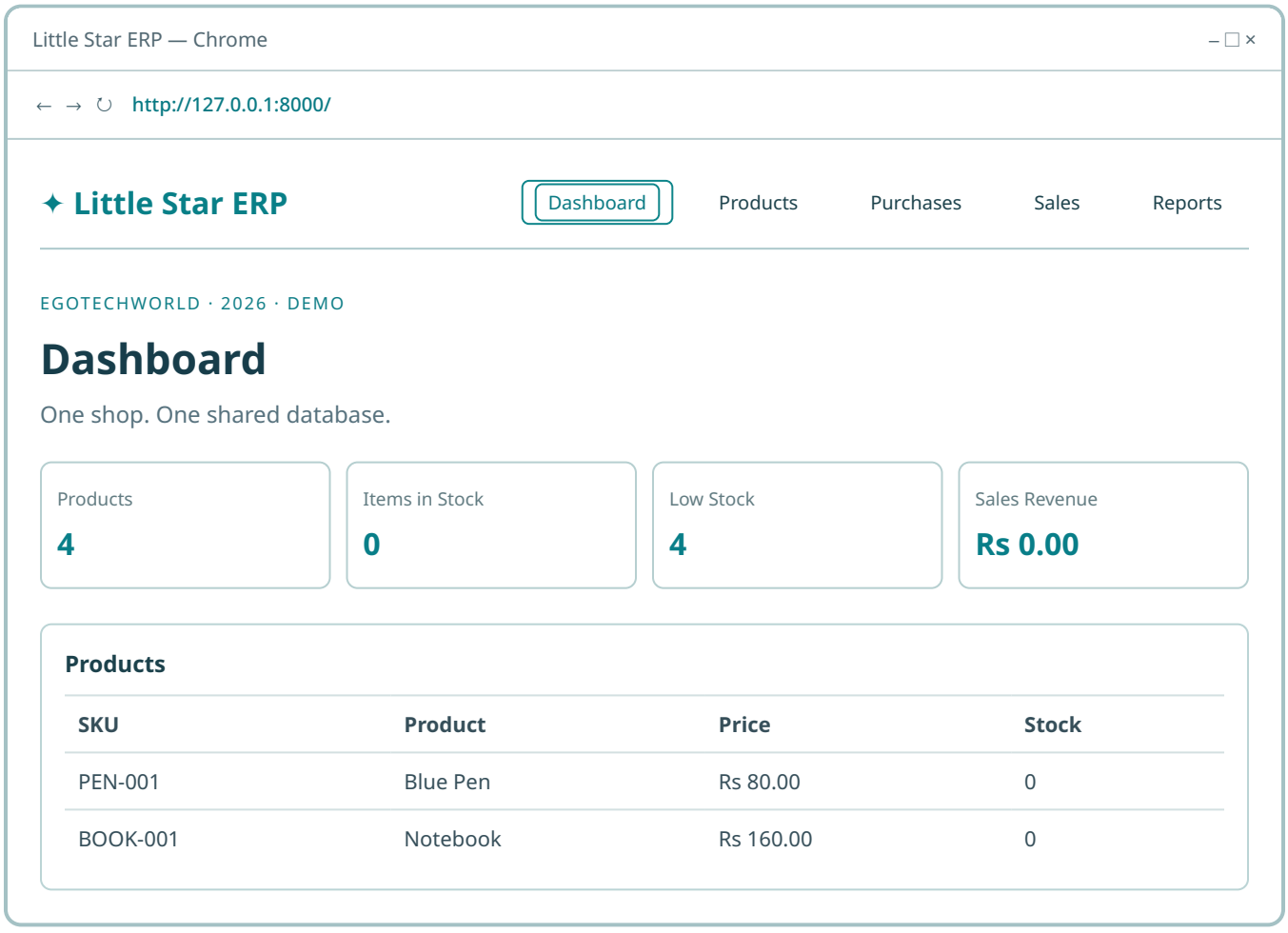
මතක තබාගන්න: Real production app එකකට strong auth, authorization, backup, encryption, logs, maintenance අවශ්‍යයි.

අපේ ERP screen එක

Dashboard එක shop එකේ වත්මන් තත්ත්වය කෙටියෙන් පෙන්වනවා.

දැනගත යුතු කරුණු

- Metric 1: total products.
- Metric 2: total items in stock.
- Metric 3: low stock; Metric 4: total sales revenue.



Illustrative Chrome-style preview using the sample app's fields; live screen may vary.

✎ ඔබේ අභ්‍යාසය

Dashboard metrics හතරේ අර්ථය ඔබේ වචනවලින් කියන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Project folder tree

ප්‍රධාන files හරියට හඳුනාගනිමු.

දැනගත යුතු කරුණු

- `app.py` = local web server + pages.
- `db.py` = data access + business rules.
- `schema.sql`, `assets/style.css`, `tests.py`, `docs/` = structure, design, testing.

VS Code · EXPLORER

```
EGOTECHWORLD_Little_Star_ERP_2026/
  |-- app.py
  |-- db.py
  |-- schema.sql
  |-- tests.py
  |-- run_windows.bat
  |-- assets/
  |    `-- style.css
  |-- docs/
  |    |-- SRS.md
  |    |-- QA_Test_Plan.md
  |    `-- Data_Dictionary.md
  `-- data/  (created when app runs)
```

✎ ඔබේ අභ්‍යාසය

Source ZIP එක extract කර tree එකට ගැළපෙන files count කරන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Database connection කේතය

SQLite connection එක object එකක්. Table rows කියවීමට සහ වෙනස් කිරීමට අපි එය භාවිතා කරනවා.

දැනගත යුතු කරුණු

- Python standard-library sqlite3 import කරන්න.
- Foreign keys enable කිරීමෙන් references පරීක්ෂා කරයි.
- Database path local file එකකට පමණයි.



> ✎ **ඔබේ අභ්‍යාසය**
>
> `db.py` file එකෙන් `sqlite3.connect` line එක සොයා බලන්න.

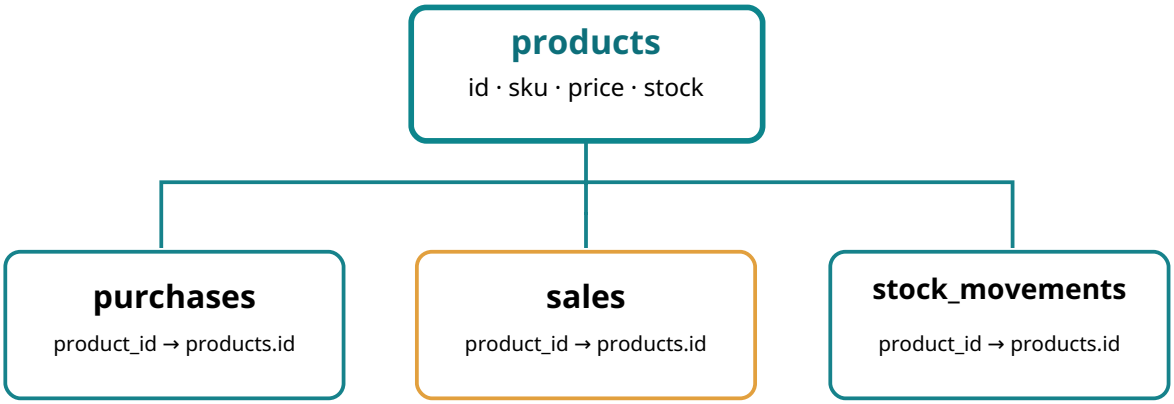
මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Database schema එක

Tables 4ක් සාදා එක් එක් table එකක අරමුණ පැහැදිලි කරමු.

දැනගත යුතු කරුණු

- Products table එක SKU + name + price + stock රඳවා ගනී.
- Purchases සහ Sales tables transactions රඳවා ගනී.
- Stock movements table එක change log එකකි.



✎ **ඔබේ අභ්‍යාසය**

schema.sql හි CREATE TABLE statements හතර සොයන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Product table එක සෑදීම

Product එකක් මුලින්ම stock zero සමඟ create වෙයි.

දැනගත යුතු කරුණු

- `id` යනු unique primary key.
- `sku` unique text code.
- `price_cents` මුදල් cents වලින්; `stock_qty` whole units වලින්.

Visual Studio Code

schema.sql

EXPLORER

```
Little-Star-ERP/
|-- app.py
|-- db.py
|-- schema.sql
|-- tests.py
`-- docs/
```

```
CREATE TABLE IF NOT EXISTS products (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  sku TEXT NOT NULL UNIQUE,
  name TEXT NOT NULL,
  price_cents INTEGER NOT NULL CHECK(price_cents > 0),
  stock_qty INTEGER NOT NULL DEFAULT 0
    CHECK(stock_qty >= 0),
  reorder_level INTEGER NOT NULL DEFAULT 3
);
```

✎ ඔබේ අභ්‍යාසය

Blue Pen සඳහා price_cents=8000 කියන්නේ Rs 80.00 වෙන්නේ කොහොමද?

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Product seed data එක

පළමු run එකේදී sample products හතරක් automatically insert වෙනවා.

දැනගත යුතු කරුණු

- Blue Pen Rs 80; Notebook Rs 160.
- Pencil Rs 45; Eraser Rs 30.
- Start stock zero නිසා Purchasing page එකෙන් first delivery එක record කරන්න.



✎ ඔබේ අභ්‍යාසය

ඔබ කැමති Ruler SKU එකක් සහ price එකක් add කරන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Python HTTP server එක

Browser එක HTTP request එකක් යවනකොට server class එක එයට ප්‍රතිචාර දෙනවා.

දැනගත යුතු කරුණු

- `GET /` = dashboard page එක.
- `GET /products` = products page එක.
- `POST /sales` = sale form data එක receive කරන action.

Visual Studio Code

app.py (concept excerpt)

EXPLORER

```
Little-Star-ERP/
|-- app.py
|-- db.py
|-- schema.sql
|-- tests.py
`-- docs/
```

```python
from http.server import BaseHTTPRequestHandler

class ERPHandler(BaseHTTPRequestHandler):
    def do_GET(self):
        # Send the requested page.
        pass

    def do_POST(self):
        # Validate and save a form.
        pass
```

✎ ඔබේ අභ්‍යාසය

GET සහ POST අතර වෙනස වාක්‍ය දෙකකින් ලියන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

127.0.0.1 කියන්නේ?

ඔබේ browser එකත් Python server එකත් එකම computer එකේ run වෙනවා.

දැනගත යුතු කරුණු

- `127.0.0.1` = local loopback address.
- `8000` = server port.
- මෙය Internet එකට deploy කිරීමක් නොවේ.

Visual Studio Code

app.py (concept excerpt)

```
EXPLORER

Little-Star-ERP/
|-- app.py
|-- db.py
|-- schema.sql
|-- tests.py
`-- docs/
```

```
# Only accept traffic from this computer.
server = ThreadingHTTPServer(
    ('127.0.0.1', 8000), ERPHandler
)
server.serve_forever()
```

> ✎ **ඔබේ අභ්‍යාසය**
>
> Browser address bar එකට `http://127.0.0.1:8000` type කරන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

HTML Page + CSS

Browser එකට HTML structure සහ CSS styling යැවූ විට screen එක ලස්සනට පෙනේ.

දැනගත යුතු කරුණු

- HTML heading, forms, tables ඉදිරිපත් කරයි.
- CSS අවශ්‍ය spacing, fonts, borders පාලනය කරයි.
- අපේ browser app එකත් සුදු background design එකක් භාවිතා කරයි.

Visual Studio Code

assets/style.css

EXPLORER

```
Little-Star-ERP/
|-- app.py
|-- db.py
|-- schema.sql
|-- tests.py
`-- docs/
```

```css
/* Clean, white browser design. */
body {
  background: #fff;
  color: #163b46;
  font-family: Arial, sans-serif;
}
```

✎ ඔබේ අභ්‍යාසය

CSS `background:#fff` යන්නෙන් අදහස් වන්නේ කුමක්ද?

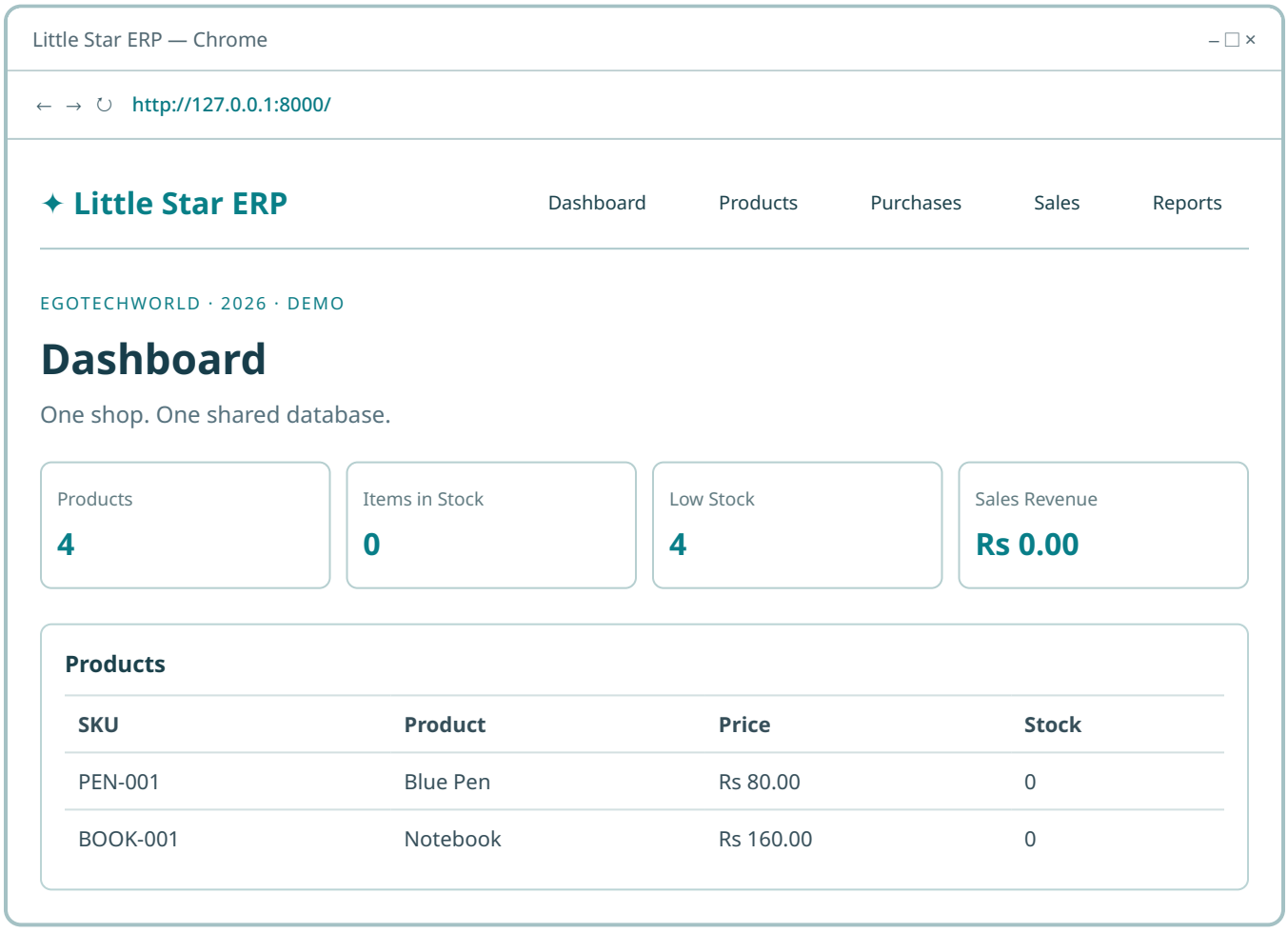
මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Navigation පිටු පහ

Menu එකෙන් modules අතර ගමන් කරමු.

දැනගත යුතු කරුණු

- Dashboard → Products → Purchasing → Sales → Reports.
- කුමන page එකක සිටියත් top navigation එක පෙනේ.
- Modules වෙනස් වුණත් database එක එකමයි.



Illustrative Chrome-style preview using the sample app's fields; live screen may vary.

✎ ඔබේ අභ්‍යාසය

Browser එකේ links පහම click කර පිටු හැරෙනවාද බලන්න.

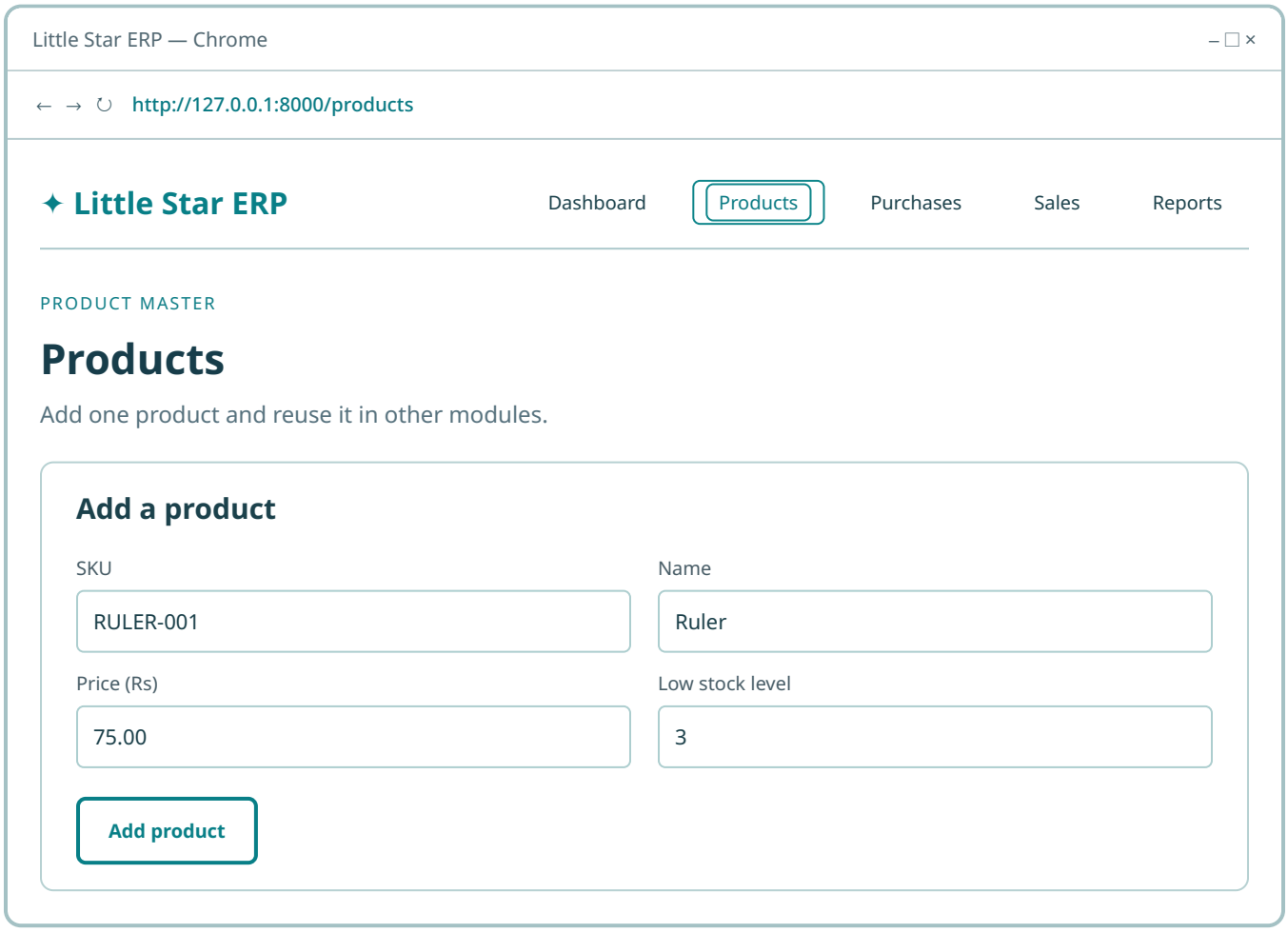
මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Products Page

Add product form එකක් සහ list table එකක්.

දැනගත යුතු කරුණු

- Input: SKU, name, selling price, reorder level.
- Save → database products table.
- Duplicate SKU දුන්නොත් friendly error.



Illustrative Chrome-style preview using the sample app's fields; live screen may vary.

✎ ඔබේ අභ්‍යාසය

RULER-001 / Ruler / 75.00 / level=3 add කරන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Price input validate කරන්න

“seventy five” වැනි text එකක් price එකක් නොවේ. Decimal method එකෙන් cents බවට පරිවර්තනය කරමු.

දැනගත යුතු කරුණු

- `Decimal("75.25") * 100` → 7525 cents.
- Negative values, zero සහ decimal digits තුනක් reject කරයි.
- Money එක raw floating-point ලෙස නොතබා integer cents වලින් store කරයි.



> ✎ **ඔබේ අභ්‍යාසය**
>
> Rs 9.99 යනු cents කීයද?

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Products insert කරන function

Product name සහ price check කර database record එකක් එක් කරන function එකට `add_product` කියනවා.

දැනගත යුතු කරුණු

- Empty name reject කරන්න.
- SKU unique rule එක SQLite පරීක්ෂා කරයි.
- නව product එක stock 0න් පටන් ගනී.



> ✎ **ඔබේ අභ්‍යාසය**
>
> SKU duplicate වුණොත් expected error message එක කියන්න.

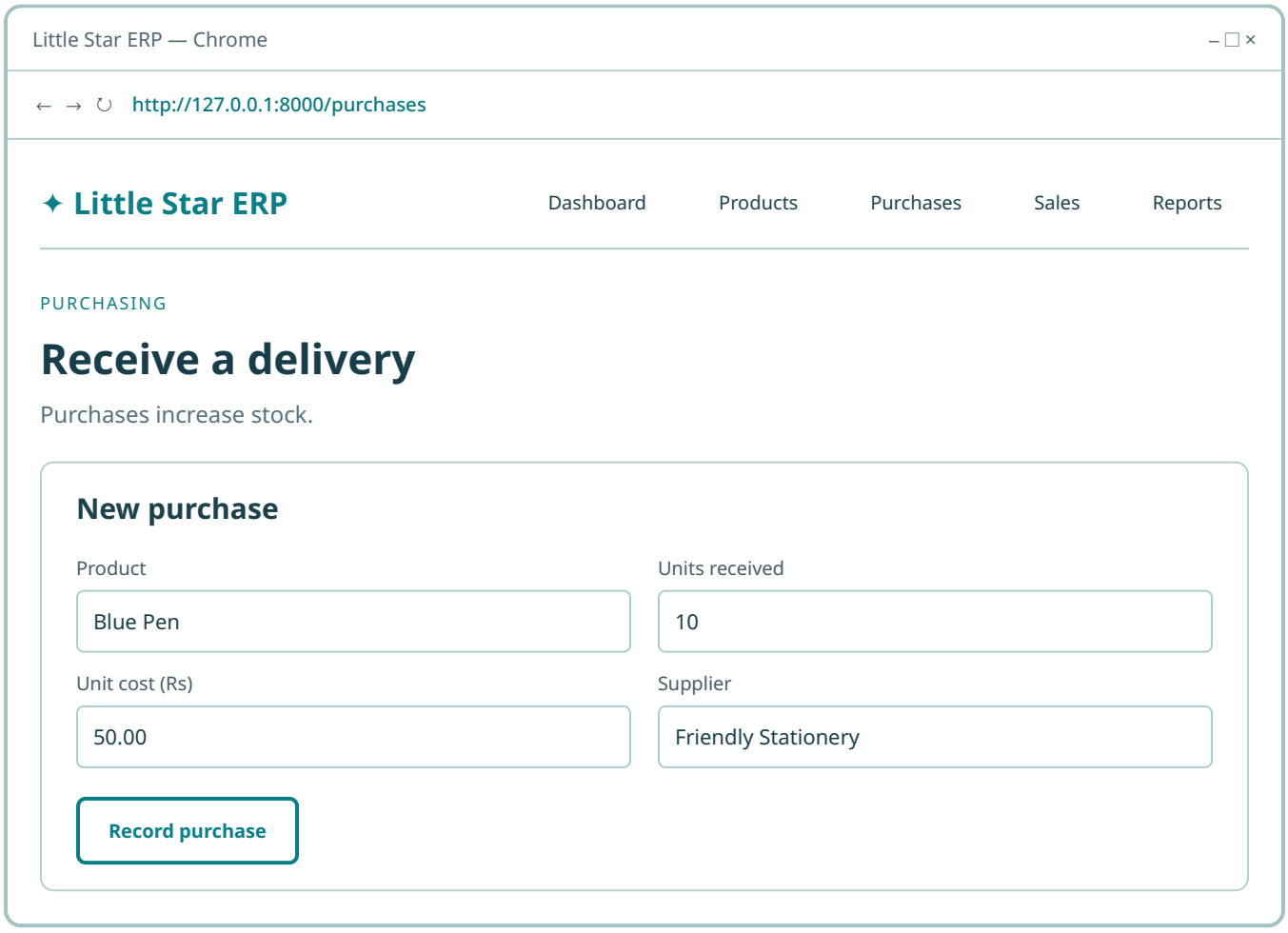
මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Purchasing Page

Supplier කෙනෙකුගෙන් ලැබෙන delivery එක record කරන්න.

දැනගත යුතු කරුණු

- Product select කරන්න.
- Quantity, unit cost, supplier type කරන්න.
- Save කර Products page එකේ stock වැඩි වූ බව බලන්න.



Illustrative Chrome-style preview using the sample app's fields; live screen may vary.

✎ ඔබේ අභ්‍යාසය

Blue Pen 10ක් / unit cost Rs 50 / supplier "Friendly" ලෙස save කරන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Purchase transaction එක

Stock increase සහ purchase record එක all-or-nothing ලෙස save කරන්න.

දැනගත යුතු කරුණු

- BEGIN IMMEDIATE → write transaction ආරම්භ කර lock ලබා ගනී.
- INSERT purchase, UPDATE stock, INSERT movement.
- COMMIT; error නම් ROLLBACK.



✎ ඔබේ අභ්‍යාසය

Steps තුනෙන් UPDATE fail වුණොත් movement record එක save විය යුතුද?

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Purchase +10 movement

Stock log එකේ වෙනස positive number එකකින් පෙන්වනවා.

දැනගත යුතු කරුණු

- Stock 0 → purchasing +10 → stock 10.
- Movements table: change_qty = +10.
- Reason: PURCHASE.

Visual Studio Code

db.py (movement excerpt)

```
EXPLORER

Little-Star-ERP/
|-- app.py
|-- db.py
|-- schema.sql
|-- tests.py
`-- docs/
```

```
# Store each stock change for tracing.
con.execute(
  "INSERT INTO stock_movements "
  "(product_id,change_qty,reason,reference_id) "
  "VALUES (?,?,?,?)",
  (product_id, qty, 'PURCHASE', purchase_id)
)
```

✎ ඔබේ අභ්‍යාසය

Purchase 3ක් තවත් add කළාම stock කීයද?

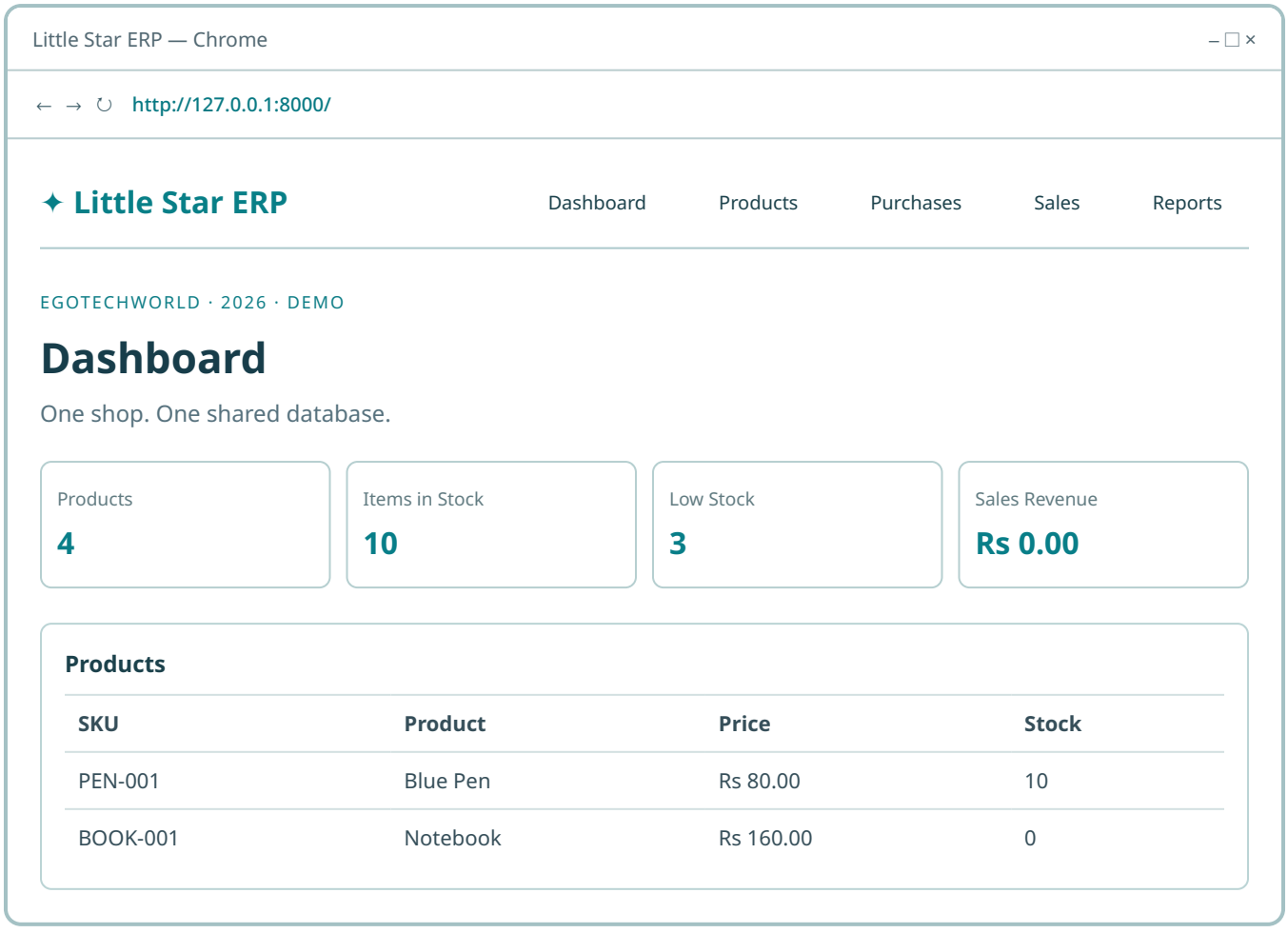
මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Products Page Refresh

Purchasing page එකෙන් dashboard වෙත ආපසු ගිය විට stock අලුත් අගයක් පෙන්විය යුතුයි.

දැනගත යුතු කරුණු

- Products table current stock කියවයි.
- Dashboard `summary()` function එක updated stock එක එකතු කරයි.
- වෙනම spreadsheet එකක් manual update කළ යුතු නැහැ.



Illustrative Chrome-style preview using the sample app's fields; live screen may vary.

✎ ඔබේ අභ්‍යාසය

Purchase කළ පසුව Browser Refresh click කරන්න. Stock නව අගයද?

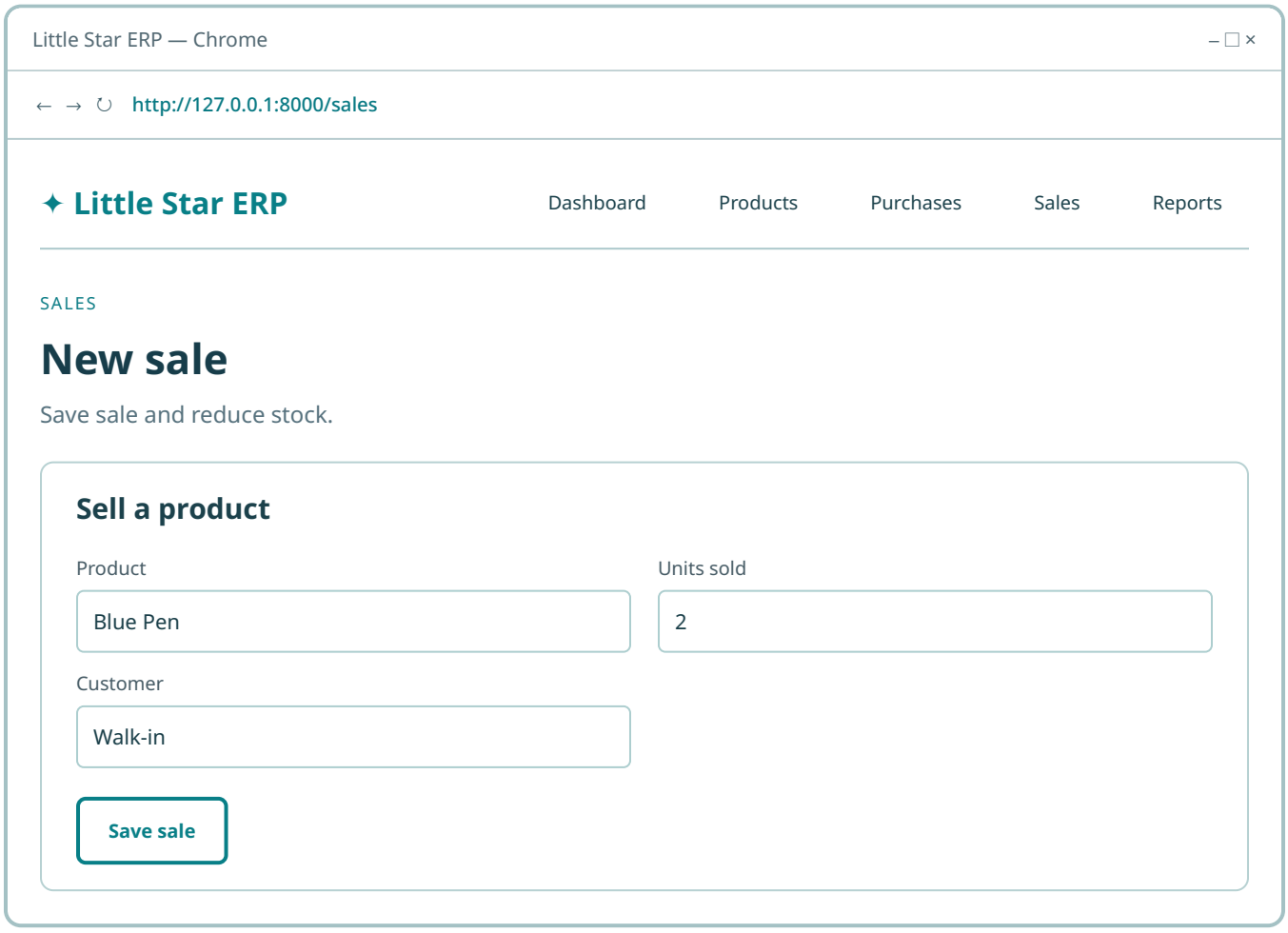
මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Sales Page

Customer කෙනෙකුට භාණ්ඩ විකුණන form එක.

දැනගත යුතු කරුණු

- Product select; units sold type.
- Customer name optional — `Walk-in` භාවිතා කරයි.
- Stock ප්‍රමාණවත් නම් sale record එක save කරයි.



Illustrative Chrome-style preview using the sample app's fields; live screen may vary.

✎ ඔබේ අභ්‍යාසය

Blue Pen දෙකක් sale කරන්න. Stock 10 → 8 බලන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Sales validation එක

Sale එකක් accepted වීම සඳහා requested units available stock වලට අඩු හෝ සමාන විය යුතුයි.

දැනගත යුතු කරුණු

- Stock 8, sell 2 → allow.
- Stock 8, sell 9 → reject.
- Zero හෝ negative quantity → reject.

Visual Studio Code

db.py (sale validation)

```
EXPLORER

Little-Star-ERP/
|-- app.py
|-- db.py
|-- schema.sql
|-- tests.py
`-- docs/
```

```python
# Prevent selling more than the shop owns.
if product['stock_qty'] < qty:
    raise ValueError('Not enough stock for this sale.')

# Only then save the sale and reduce stock.
```

> ✎ **ඔබේ අභ්‍යාසය**
>
> Stock=4, request=5 කියලා input test එකක් ලියන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Revenue calculate කිරීම

Selling price × quantity = sales revenue.

දැනගත යුතු කරුණු

- Blue Pen selling price 8000 cents (Rs 80).
- Sell 2 → total 16000 cents (Rs 160).
- Store sale-time price; future price changes do not rewrite old transactions.

Visual Studio Code

sales_math.py

EXPLORER

Little-Star-ERP/
|-- app.py
|-- db.py
|-- schema.sql
|-- tests.py
`-- docs/

```
# Use integer cents for monetary arithmetic.
unit_price_cents = 8000  # Rs 80
quantity = 2
total_cents = unit_price_cents * quantity
print(total_cents)  # 16000
```

✎ ඔබේ අභ්‍යාසය

Rs 160 × 3 = කීයද? Cents value එකත් ලියන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Sales Transaction 3 Steps

මිතුරන් දෙන්නෙක් එකවර sale කරන්න උත්සාහ කළත් stock data inconsistent නොවීම වැදගත්.

දැනගත යුතු කරුණු

- BEGIN IMMEDIATE; get current stock.
- Check qty, INSERT sale, deduct stock.
- INSERT movement (-qty), COMMIT.



✎ ඔබේ අභ්‍යාසය

Transaction අතර error ආවොත් stock එක පරණ value එකට යා යුත්තේ ඇයි?

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Stock -2 movement

Sale එකකට stock change negative අගයක්.

දැනගත යුතු කරුණු

- Sale 2 units → stock movement -2.
- Purchase +10 සහ sale -2 එකතුව = 8.
- Report එකේ change_qty signs බැලූවිට ගමන හඳුනාගන්න පුළුවන්.

Visual Studio Code

db.py (sale movement)

EXPLORER

Little-Star-ERP/
|-- app.py
|-- db.py
|-- schema.sql
|-- tests.py
`-- docs/

```
# Negative stock movement records units sold.
con.execute(
  "INSERT INTO stock_movements "
  "(product_id,change_qty,reason,reference_id) "
  "VALUES (?,?,'SALE',?)",
  (product_id, -qty, sale_id)
)
```

✎ ඔබේ අභ්‍යාසය

Purchase +5, sale -1, sale -2 → stock balance කීයද?

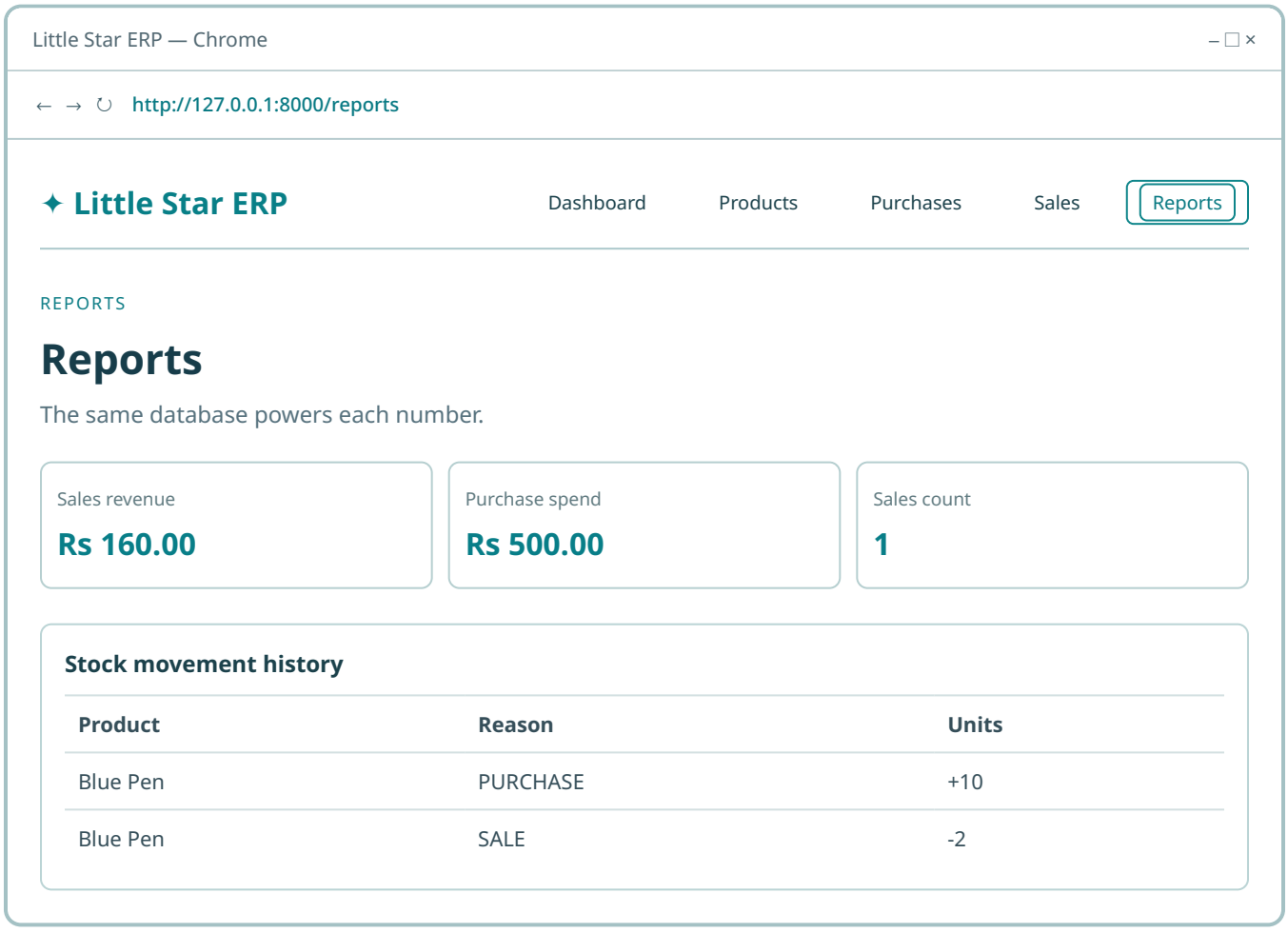
මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Reports Page

අපේ data වලින් manager ට වැදගත් සාරාංශයක් පෙන්වමු.

දැනගත යුතු කරුණු

- Sales revenue = `SUM(total_cents)`.
- Purchase spend = `SUM(qty * unit_cost_cents)`.
- Recent transactions හා stock movement history.



Illustrative Chrome-style preview using the sample app's fields; live screen may vary.

✎ ඔබේ අභ්‍යාසය

Reports එකේ Sales revenue Rs 160 පෙනෙනවාද බලන්න.

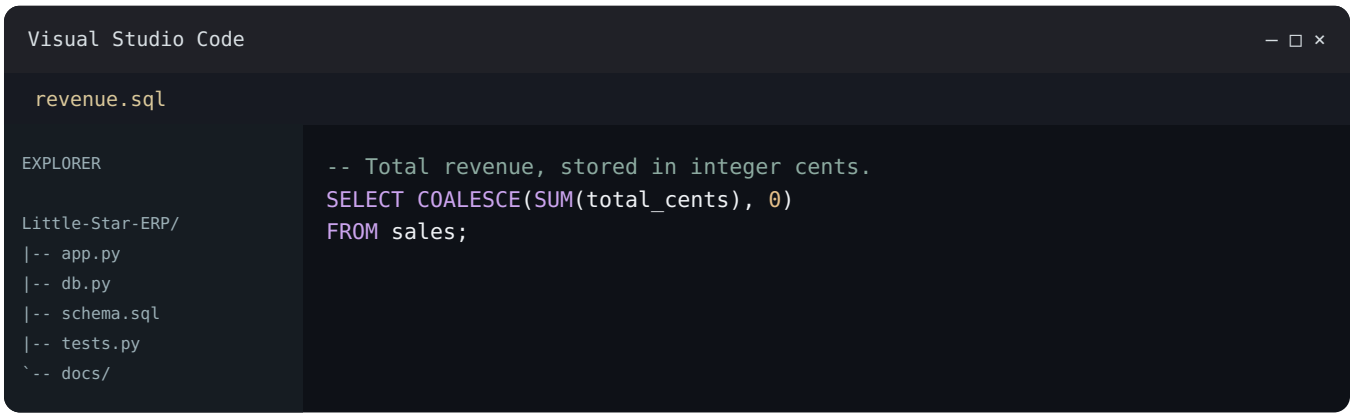
මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

SQL SUM භාවිතය

Sale records කිහිපයක එකතුව ගණනය කිරීමට SQL aggregation භාවිතා කරමු.

දැනගත යුතු කරුණු

- SUM(total_cents) = සියලු sale totals එකතුව.
- COALESCE(...,0) = rows නැතිවුණත් 0 ලබාදේ.
- Group by product වැනි reports future lesson එකක්.



> ✎ **ඔබේ අභ්‍යාසය**
>
> Sale Rs 160, Rs 90, Rs 200 නම් total revenue කීයද?

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Stock Alerts – Low Stock

ඇති භාණ්ඩ ගණන තීරණය කළ reorder level එකට අඩු වුණාම manager ට warning.

දැනගත යුතු කරුණු

- Products table එකෙන් low-stock count ගන්නවා.
- Stock qty = reorder level නම් warning පෙනේ.
- Reorder level shop එකට ගැළපෙන ලෙස තෝරාගන්න.



✎ ඔබේ අභ්‍යාසය

Notebook reorder level 4, stock 3 නම් alert එක පෙනෙනවාද?

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Server errors හඳුනාගන්න

Project එක run නොවුණොත් error message එකේ වැදගත් line එක හොයන්න.

දැනගත යුතු කරුණු

- File not found → correct folder එකේ සිටිනවාද බලන්න.
- Address already in use → port 8000 භාවිතා කරන app එකක්ද?
- Python missing → installation/PATH පරීක්ෂා කරන්න.

```
Windows Command Prompt                                    _ □ ×

C:\Users\Student> cd Desktop\EGOTECHWORLD_Little_Star_ERP_2026
C:\Users\Student> dir
C:\Users\Student> py app.py
```

> ✎ **ඔබේ අභ්‍යාසය**
>
> “No such file app.py” කියලා ආවොත් first check එක කුමක්ද?

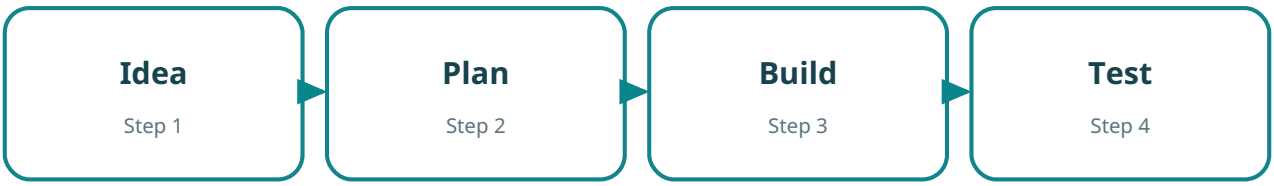
මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

End-to-end Demo

එක user journey එකෙන් modules හතරම සම්බන්ධ වෙන හැටි පරීක්ෂා කරමු.

දැනගත යුතු කරුණු

- Product “Blue Pen” දැනටමත් ඇත.
- Purchasing: 10 units at Rs 50; Sales: 2 units.
- Products: stock 8; Reports: revenue Rs 160; movement +10, -2.



> ✎ **ඔබේ අභ්‍යාසය**
>
> මේ workflow එක notebook එකේ tick boxes හතරක් සමඟ ලියන්න.

මතක තබාගන්න: මෙය අවසාන project acceptance test එකේ මූලික scenario එකකි.

Offline use සහ Restart

අපේ project එක local-only නිසා internet නැතිව භාවිතා කළ හැක.

දැනගත යුතු කරුණු

- Browser CSS local file එකක්; CDN download එකක් අවශ්‍ය නැහැ.
- SQLite `.sqlite3` file එක restart කළත් තොරතුරු රඳවා ගනී.
- New test run එකකට වෙන database copy එකක් භාවිතා කරන්න.



✎ ඔබේ අභ්‍යාසය

Server stop කර නැවත start කර ඔබගේ stock values තවම පවතිනවාද බලන්න.

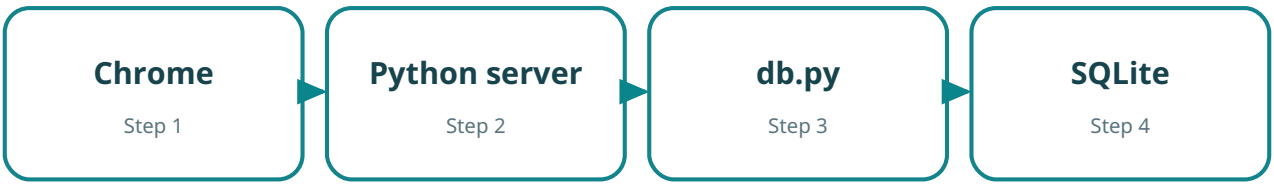
මතක තබාගන්න: Database backup එකක් නැතිව `data/` folder එක මකා දමන්න එපා.

Code reader exercise

`app.py` සහ `db.py` සම්බන්ධ වෙන lines කිහිපයක් සොයා code reading කරන්න.

දැනගත යුතු කරුණු

- app.py imports `db`.
- Product form data `db.add_product` වෙත යයි.
- Sale form data `db.sell_stock` වෙත යයි.



> ✎ **ඔබේ අභ්‍යාසය**
>
> Code එක නොවෙනස් කර `sell_stock` definition එක සොයා notebook එකේ line එකේ අදහස ලියන්න.

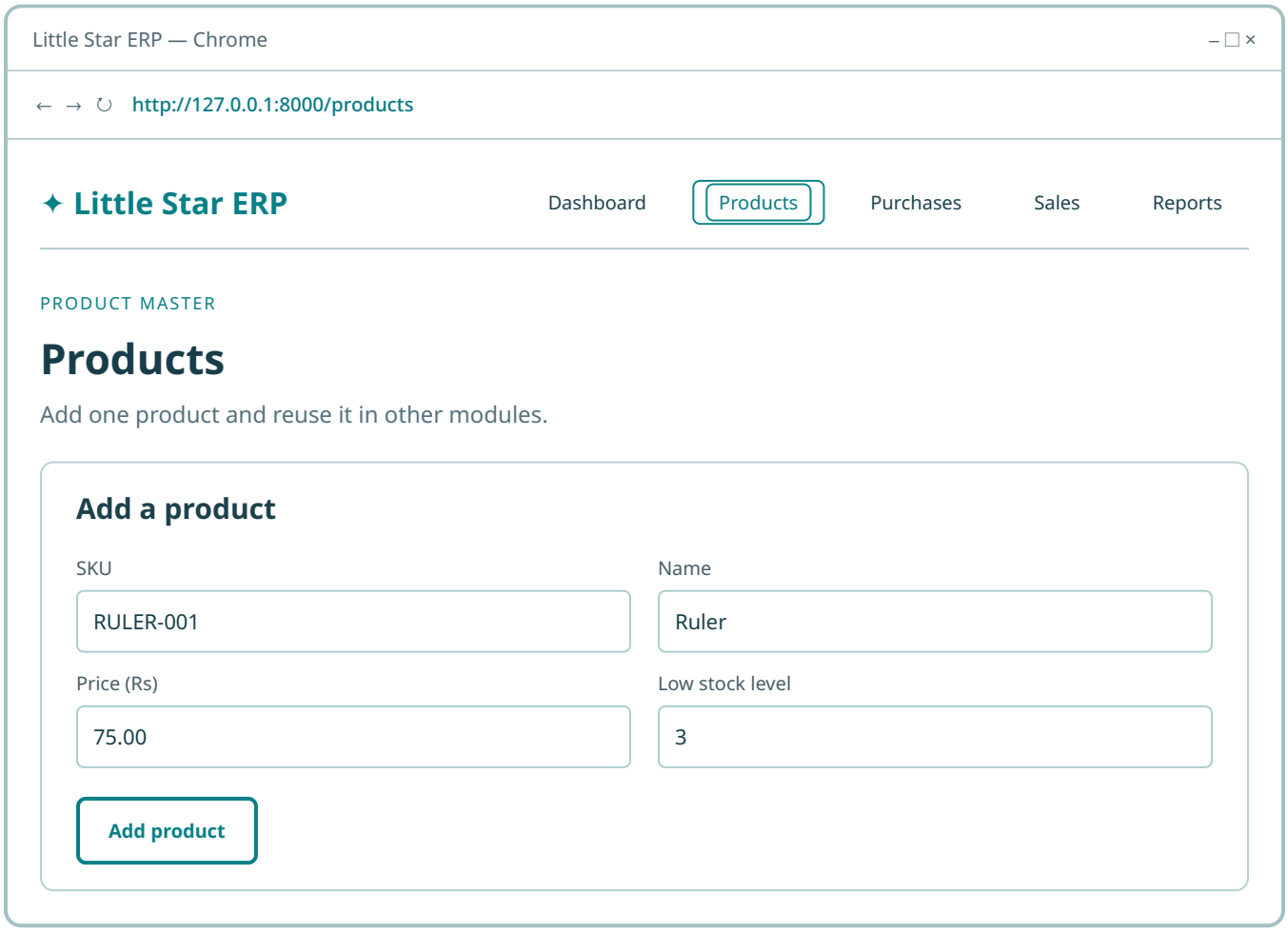
මතක තබාගන්න: Code කියවීම developer කෙනෙකුගේ ඉතා වැදගත් හැකියාවක්.

Project අලුත් product එකක්

Blue Pen පමණක් නොව Ruler එකක් එකතු කර workflows පරීක්ෂා කරන්න.

දැනගත යුතු කරුණු

- Products page: SKU=RULER-001, Name=Ruler, Price=75.00.
- Purchasing page: units=5, cost=42.00.
- Sales page: Ruler 1ක් විකුණා balance 4 සහ report value බලන්න.



Illustrative Chrome-style preview using the sample app's fields; live screen may vary.

✎ ඔබේ අභ්‍යාසය

Ruler stock හා sales revenue math එක කඩදාසියේ ලියන්න.

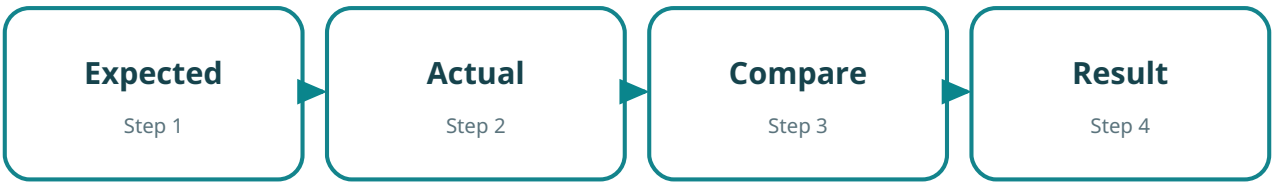
මතක තබාගන්න: Tests repeat කිරීම regression testing එකේ පියවරකි.

Testing කියන්නේ මොකක්ද?

Software එකක “වැඩ කරනවා” කියන කතාව verify කරන ක්‍රමය testing.

දැනගත යුතු කරුණු

- Expected result එකක් ලියන්න.
- Actual result එක run කර බලන්න.
- Expected != Actual නම් bug investigate කරන්න.



✎ ඔබේ අභ්‍යාසය

Stock 10, sale 2 සඳහා expected සහ actual values ටික ලියන්න.

මතක තබාගන්න: Testing යනු code fail කිරීම නොව confidence ඇති කිරීම.

Unit Tests

එක function එකක් තනිව test කිරීම unit testing.

දැනගත යුතු කරුණු

- `parse_money("75.25")` → 7525.
- `positive_int("3")` → 3.
- Invalid negative quantities reject විය යුතුයි.

Visual Studio Code

tests.py (excerpt)

```
EXPLORER

Little-Star-ERP/
|-- app.py
|-- db.py
|-- schema.sql
|-- tests.py
`-- docs/
```

```python
import unittest
import app

class MoneyTests(unittest.TestCase):
    def test_money_cents(self):
        self.assertEqual(app.parse_money('75.25'), 7525)
```

> ✎ **ඔබේ අභ්‍යාසය**
>
> `parse_money("7.001")` error විය යුත්තේ ඇයි?

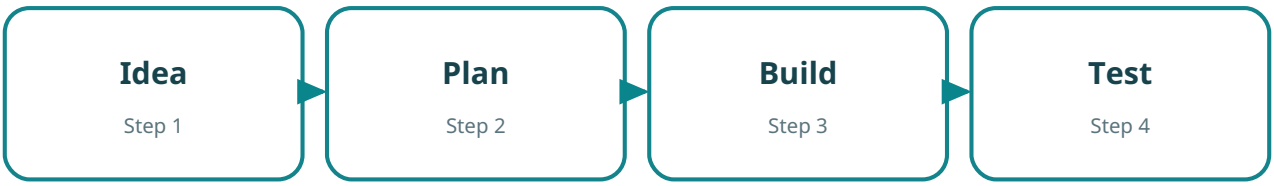
මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Integration Testing

Module දෙකක් හෝ වැඩි ගණනක් එකට වැඩ කරනවාද බලමු.

දැනගත යුතු කරුණු

- Purchasing + Inventory → stock increases.
- Sales + Inventory → stock decreases.
- Reports + tables → correct summary.



✎ ඔබේ අභ්‍යාසය

Purchase+Sale+Report test එකක් steps තුනකින් ලියන්න.

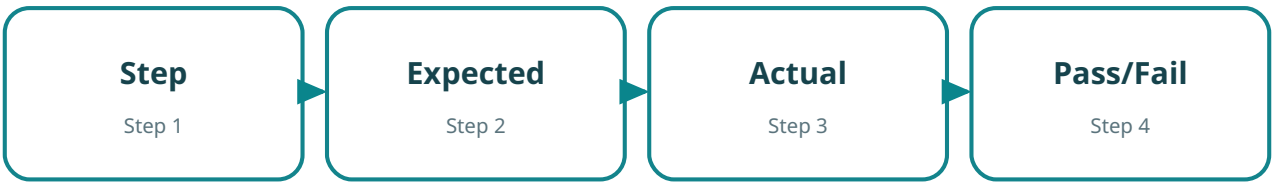
මතක තබාගන්න: Real business value එක modules එකට සම්බන්ධ වීමෙන් ලැබේ.

Manual Test Cases

QA Tester කෙනෙකුට නැවත පරීක්ෂා කළ හැකි steps.

දැනගත යුතු කරුණු

- TC-01: local app open → dashboard.
- TC-04: buy 10 → stock 10.
- TC-05: sell 2 → stock 8, revenue 160.



> ✎ **ඔබේ අභ්‍යාසය**
>
> Test Case ID, Steps, Expected, Actual, Pass/Fail columns හතරක් හදාගන්න.

මතක තබාගන්න: Source ZIP එකේ `docs/QA_Test_Plan.md` ඇත.

Negative Tests

Invalid inputs දුන්නාම app එක crash නොවී safe response ලබා දිය යුතුයි.

දැනගත යුතු කරුණු

- Quantity = 0 → reject.
- Price = -5 → reject.
- Request sale > available stock → reject.



✎ ඔබේ අභ්‍යාසය

ඔබට හිතෙන තවත් bad input තුනක් ලියන්න.

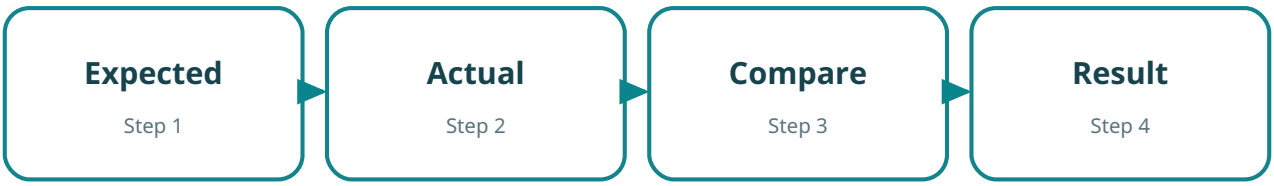
මතක තබාගන්න: “Happy path” එකක් හොඳයි; unhappy paths ද test කරන්න.

Boundary Tests

Rules වල මධ්‍යයේ පමණක් නොව අසල values වලද bugs තිබිය හැක.

දැනගත යුතු කරුණු

- Stock=5 → sale 5 accepted.
- Stock=5 → sale 6 rejected.
- Quantity min=1; quantity 0 rejected.



✎ **ඔබේ අභ්‍යාසය**

Allowed quantity 100000 නම් 100001 value එක ගැන expected result ලියන්න.

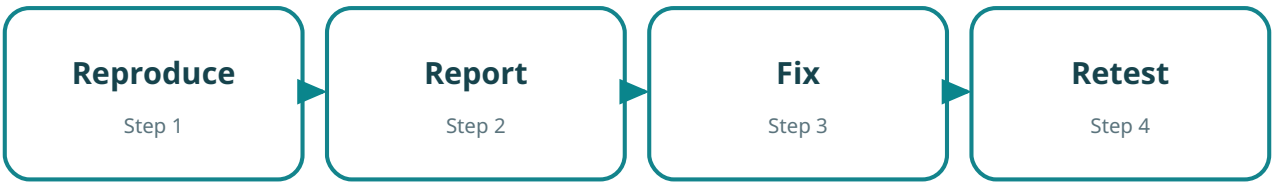
මතක තබාගන්න: Just below / exactly / just above methodology එකක් භාවිතා කරන්න.

Bug report ලියමු

Bug එකක් සොයාගත්තාම developer ට නැවත ඒක ඇතිකරන්න පුළුවන් විදිහට ලියන්න.

දැනගත යුතු කරුණු

- Title: “Oversell unexpectedly allowed”.
- Steps: start stock 2 → request sale 5 → click Save.
- Expected: error. Actual: stock -3 නම් serious bug.



✎ ඔබේ අභ්‍යාසය

Severity=High වන්නේ ඇයි කියලා වාක්‍යයක් ලියන්න.

මතක තබාගන්න: Screenshot, environment, exact steps, actual/expected තිබුණොත් fix කිරීමට පහසුයි.

18 automated tests

Source ZIP එකේ tests 18ක් අන්තර්ගතයි. අපි ඒවා Python standard library එකෙන් run කරමු.

දැනගත යුතු කරුණු

- `py -m unittest tests.py -v` command එක run කරන්න.
- Test names එකින් එක status “ok” පෙන්වයි.
- Fail එකක් තිබුණොත් code/expected values inspect කරන්න.

```
Windows Command Prompt                                  _ □ ×

C:\Users\Student> py -m unittest tests.py -v
```

> ✎ **ඔබේ අභ්‍යාසය**
>
> ඔබේ computer එකේ passed tests ගණන සටහන් කරන්න.

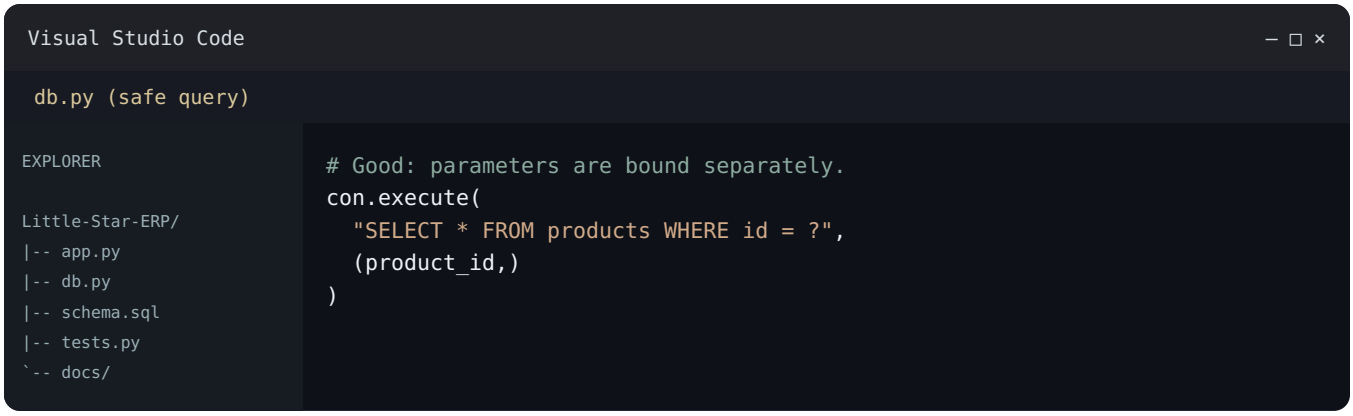
මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

SQL Injection කියන්නේ?

User text එක query එකේ command කොටසක් ලෙස concatenate කළොත් අවදානමක් ඇතිවේ.

දැනගත යුතු කරුණු

- Prepared parameter placeholders `?` භාවිතා කරන්න.
- SQLite driver එක user data ලෙස values safely bind කරයි.
- Database permissions හා validation තවත් වැදගත්.



> ✎ **ඔබේ අභ්‍යාසය**
>
> Input value එක query string එකට අතින් එකතු නොකරන්නේ ඇයි?

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

HTML Escaping

Customer name එකට text යොදා browser එකට dangerous HTML යැවීම වැළැක්වීම.

දැනගත යුතු කරුණු

- User text output කරනකොට `html.escape()` භාවිතා කරමු.
- `<script>` වැනි input code ලෙස run නොවී text ලෙස පෙන්විය යුතුයි.
- Input validation + output escaping එකිනෙක වෙනස් duties.



✎ ඔබේ අභ්‍යාසය

HTML text සහ HTML instruction අතර වෙනස කියන්න.

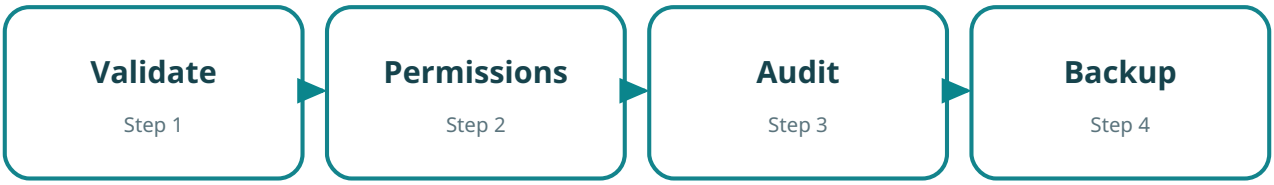
මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

CSRF form token

වෙනත් website එකකින් local app form එකකට unwanted POST එකක් දීමේ අවදානම අඩු කිරීමට token යොදාගනී.

දැනගත යුතු කරුණු

- Form එක hidden CSRF token එක යවයි.
- Server එක submitted token verify කරයි.
- Demo implementation process-memory session store භාවිතා කරයි; real deployment සඳහා පූර්ණ security design අවශ්‍යයි.



✎ ඔබේ අභ්‍යාසය

Token missing නම් server එක කුමක් කළ යුතුද?

මතක තබාගන්න: CSRF protection එක alone authentication/authorization වෙනුවට නොවේ.

Database Transaction QA

Sale save වුණාත් stock update fail වුණොත් report සහ stock inconsistent වේ.

දැනගත යුතු කරුණු

- All changes COMMIT or ROLLBACK together.
- Insufficient stock request එකක් sale row එකක් create නොකළ යුතුයි.
- Automated tests count row changes පරීක්ෂා කරයි.



✎ **ඔබේ අභ්‍යාසය**

Sale 5 reject වූ පසු stock හා sale_count වෙනස් නොවන බව test කරන්න.

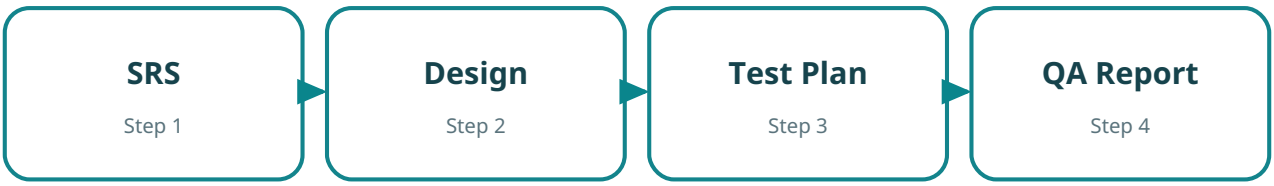
මතක තබාගන්න: Correct inventory transaction handling vital.

QA Report එකක්

Testing අවසන් වූ පසු status, tested scope, issues සහ remaining risks ලියා තබන්න.

දැනගත යුතු කරුණු

- Test date, environment, test count, pass/fail.
- Known limitations: no login, no backups, no accounting ledger.
- Go/No-Go: local learning demo පමණක් use.



✎ ඔබේ අභ්‍යාසය

Release note එකක “Do not deploy publicly” කියලා සඳහන් කරන්න.

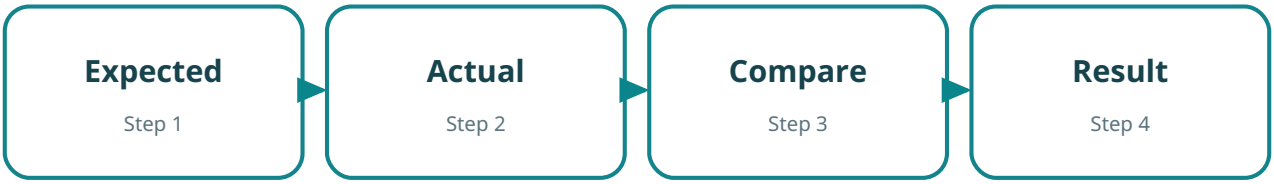
මතක තබාගන්න: Automated tests passed වුණත් production readiness කියන්නේ නැහැ.

Regression Testing

අලුත් product feature එකක් change කළාම පරණ Sales feature එක තවම වැඩ කරනවාද බලන්න.

දැනගත යුතු කරුණු

- Code change එකට කලින් සහ පසුව tests run කරන්න.
- Purchase + Sale + Report sample scenario නැවත කර බලන්න.
- එක bug fix එකක් වෙන තැනකට බලපානවාද කියලා හොයන්න.



✎ ඔබේ අභ්‍යාසය

Ruler product එක add කළ පසු Pen transaction tests නැවත run කරන්න.

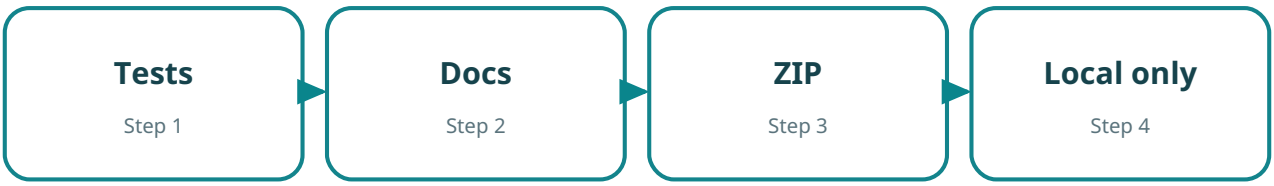
මතක තබාගන්න: Tests repeat කිරීමෙන් quality confidence වැඩිවේ.

Release Checklist

ZIP එක share කිරීමට පෙර steps එකින් එක confirm කරන්න.

දැනගත යුතු කරුණු

- README instructions clear; source files present.
- Tests passing; sample data fictional.
- No real passwords, payment data or private customer details.



✎ ඔබේ අභ්‍යාසය

Checklist එකට “backup plan” item එකක් එක් කරන්න.

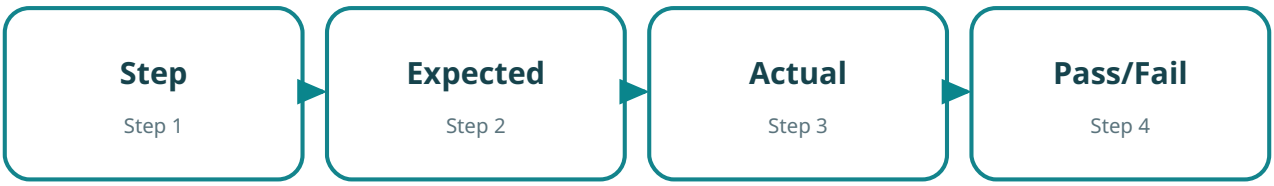
මතක තබාගන්න: Local development app එක public website එකකට upload නොකරන්න.

Student QA Mini Challenge

ඔබ tester නම් bug තුනක් හොයාගන්න පුළුවන්ද?

දැනගත යුතු කරුණු

- Test A: same SKU twice.
- Test B: stock 0දී sale 1.
- Test C: 7.001 rupees price.



✎ ඔබේ අභ්‍යාසය

Each case සඳහා Expected/Actual/Pass table එකක් පුරවන්න.

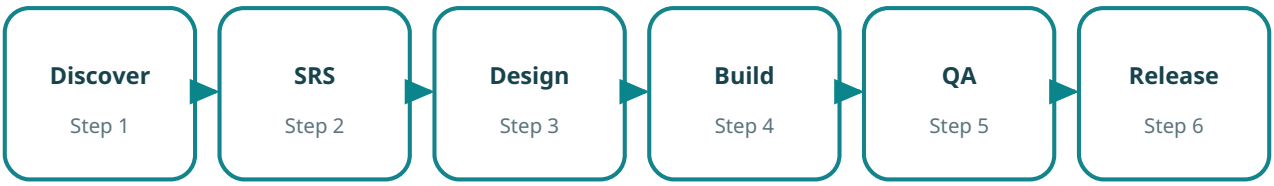
මතක තබාගන්න: QA result එක report කරන විට guess නොකර evidence දක්වන්න.

ERP implementation එකක් ආරම්භ කිරීම

Real ERP ව්‍යාපෘතියක software එක පමණක් නොව මිනිසුන් සහ ක්‍රියාදාමත් වැදගත්.

දැනගත යුතු කරුණු

- Identify pain points සහ measurable goals.
- Clean product/customer/supplier data.
- Pilot test → staff training → phased rollout.



✎ ඔබේ අභ්‍යාසය

School canteen ERP එකක් සඳහා pilot group එකක් තෝරන්න.

මතක තබාගන්න: System change management සහ adoption වැදගත්.

Roles & Permissions

සෑම සේවකයෙකුටම සෑම action එකක්ම කළ නොහැකි විය යුතුයි.

දැනගත යුතු කරුණු

- Cashier: sales record; Store keeper: receive stock.
- Manager: reports, approvals, overrides.
- Real ERP: identity, least privilege, audit, MFA when suitable.



✎ ඔබේ අභ්‍යාසය

Cashier කෙනෙකුට purchase price වෙනස් කිරීමේ අවසර අවශ්‍යද?

මතක තබාගන්න: අපේ mini ERP එකේ login/permissions නැති නිසා public use නොකළ යුතුයි.

Backups & Recovery

Database file එක නැතිවුණොත් app එකේ recorded data අහිමි විය හැක.

දැනගත යුතු කරුණු

- Routine backup schedule තෝරන්න.
- Restore test එකක් කරන්න — backup ඇත්තටම ක්‍රියාකරනවාද?
- SQLite live database backup එකට correct online backup method භාවිතා කරන්න; random copy අවදානම්.



✎ ඔබේ අභ්‍යාසය

Backup එකක් restore නොකර තිබුණත් “tested” කියලා ලියන්න පුළුවන්ද?

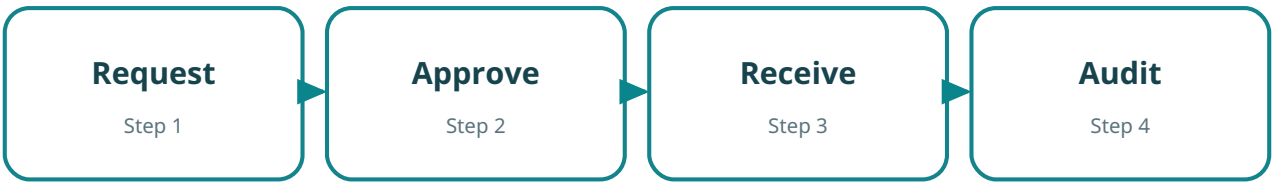
මතක තබාගන්න: Learning project ZIP එකේ live data file එක ඉවත් කර share කරන්න.

Business process approvals

සමහර වැඩ manager approve කරලා පමණක් ඉදිරියට යා යුතුයි.

දැනගත යුතු කරුණු

- Purchase request → Manager approval → Order → Goods receipt.
- අපේ learning demo එක purchase request සහ approval phases නොමැතිව receipt-only simulates කරයි.
- Approval + audit logs fraudulent changes වැළැක්වීමට උපකාරී වේ.



✎ ඔබේ අභ්‍යාසය

Rs 100,000 purchase එකක් අනුමත කරන්නේ කවුද කියලා policy එකක් යෝජනා කරන්න.

මතක තබාගන්න: Real workflow එක business rules අනුව වෙනස් වේ.

2026 Cloud ERP

Cloud deployment සහ online integrations වඩා පරිමාණවත් විය හැකි වුවත් risks පාලනය කරන්න.

දැනගත යුතු කරුණු

- Shared API connections, multi-location access, scheduled reports.
- Network outages, access control, data residency, vendor choice.
- Test + monitoring + backup + disaster recovery සැලසුම් අවශ්‍යයි.



✎ ඔබේ අභ්‍යාසය

Local ERP සහ Cloud ERP එකක් අතර advantages දෙකක් සසඳන්න.

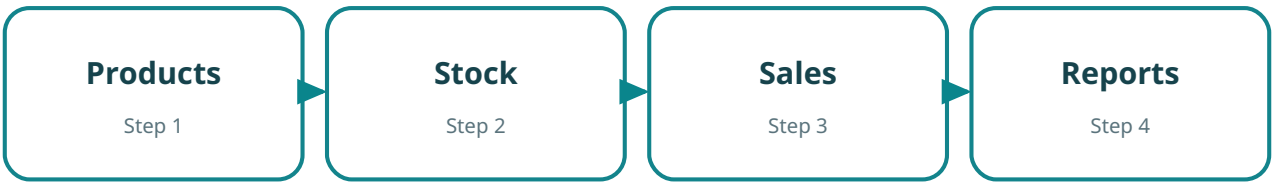
මතක තබාගන්න: Cloud කියන්නේ data magically safe කියන එක නොවේ.

ERP + Data Analytics

ERP data වලින් charts, dashboards සහ patterns සොයන්න පුළුවන්.

දැනගත යුතු කරුණු

- Best-selling products අනුව stock planning.
- Low-stock count time trend.
- Revenue trend, stock turnover වගේ KPIs අවශ්‍ය definitions සමඟ.



✎ ඔබේ අභ්‍යාසය

Week එකක pen sales quantities [3,4,5,2] එකතුව සොයන්න.

මතක තබාගන්න: Privacy හා data quality නිසා වැරදි chart conclusions වලින් වළකින්න.

AI forecasting: ඉගෙනීමට පමණයි

AI model එකක් past sales අනුව next-week demand අනුමාන කළ හැක.

දැනගත යුතු කරුණු

- Historical data අවශ්‍යයි; bad data → bad prediction.
- Seasonality (school term) නිසා trend වෙනස් විය හැක.
- Human review සහ confidence limits නැති forecast blindly භාවිතා නොකරන්න.



✎ ඔබේ අභ්‍යාසය

School reopen වීමකදී pens demand වැඩි විය හැක්කේ ඇයි?

මතක තබාගන්න: 2026 AI tools ERP support කළත් reliable operations rules වෙනස් වන්නේ නැහැ.

Integration සහ API

බාහිර software එකක් ERP data සමඟ safe ලෙස කතා කිරීමට API භාවිතා කළ හැක.

දැනගත යුතු කරුණු

- POS sale → ERP inventory update.
- E-commerce order → stock availability check.
- Auth, rate limits, validation, retries වැදගත්.



✎ ඔබේ අභ්‍යාසය

Online shop එකේ stock=0 යන විට listing “out of stock” විය යුත්තේ ඇයි?

මතක තබාගන්න: අපේ basic project එකට public API එකක් නොමැත.

ERP හි AI ගැන අතිශයෝක්ති නොකරමු

“AI enabled” කියලා ලේබල් තිබුණා කියලා result නිවැරදි බවට සහතිකයක් නැහැ.

දැනගත යුතු කරුණු

- Explain where predictions come from.
- Review permissions before AI reads company data.
- Human approve financial/stock-changing actions.



✎ ඔබේ අභ්‍යාසය

AI එක “buy 500 items now” කියලා කිව්වොත් check කරන facts තුනක් ලියන්න.

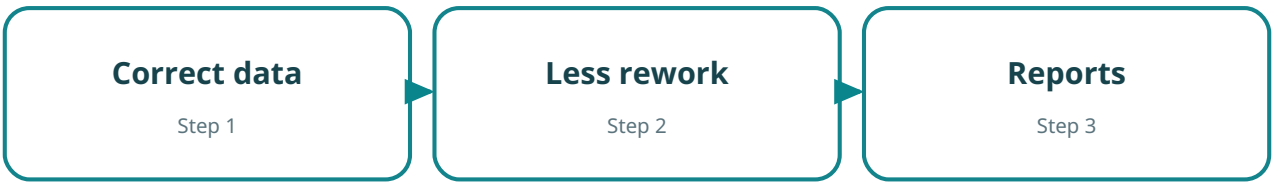
මතක තබාගන්න: AI outputs inspect කරන්න; payment/stock decisions proof නොමැතිව automate නොකරන්න.

Real ERP එකක් තෝරන්නේ කොහොමද?

ආයතනයක අවශ්‍යතාව, මුදල, සැලසුම සහ team skills එකට බලන්න.

දැනගත යුතු කරුණු

- Required modules + integrations first.
- Total cost: license, setup, training, support, migrations.
- Security, vendor support, data export, uptime සහ compliance බලන්න.



✎ ඔබේ අභ්‍යාසය

Small shop සඳහා must-have features තුනක් තෝරන්න.

මතක තබාගන්න: “Biggest ERP” එක හැම විටම best fit නොවේ.

ERP vocabulary review

ERP අලුත් වචන කීපයක් මතක් කරමු.

දැනගත යුතු කරුණු

- SKU, Inventory, Supplier, Customer, Revenue.
- Module, Database, Transaction, Primary Key.
- SRS, SDLC, QA, Regression Test, Backup.



✎ ඔබේ අභ්‍යාසය

මෙම වචන පහක අර්ථය ඔබේම වචනවලින් පැහැදිලි කරන්න.

මතක තබාගන්න: වචනයේ අර්ථය examples සමඟ සම්බන්ධ කළ විට මතක තබාගැනීම පහසුයි.

Quick Quiz 01 – ERP

නිවැරදි පිළිතුර තෝරාගන්න. ඉන්පසු හේතුව කියන්න.

දැනගත යුතු කරුණු

- 1. ERP = (A) Enterprise Resource Planning (B) Easy Robot Program.
- 2. Stock වැඩි කරන්නේ (A) Sales (B) Purchasing.
- 3. Sales revenue = (A) Profit (B) Sales amount.



✎ ඔබේ අභ්‍යාසය

Answer key: 1 A, 2 B, 3 B. හේතු තුනක් ලියන්න.

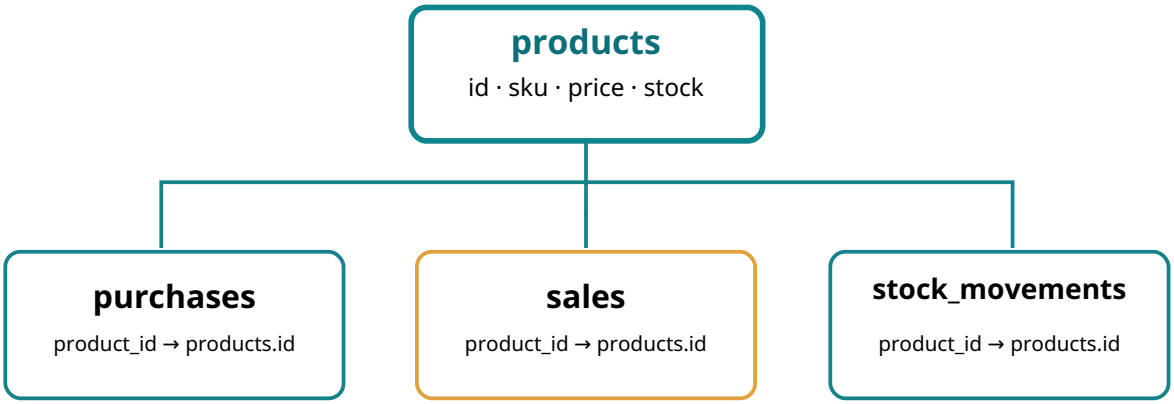
මතක තබාගන්න: ලකුණු පමණක් නොව හේතුව පැහැදිලි කරන්න.

Quick Quiz 02 – Database

SQL සහ tables ගැන මතකද බලමු.

දැනගත යුතු කරුණු

- 1. Product identifier = (A) SKU (B) Background image.
- 2. One database තුළ (A) separate truth (B) shared records.
- 3. Foreign Key භාවිතා කරන්නේ (A) link tables (B) draw logo.



✎ ඔබේ අභ්‍යාසය

Answer key: 1 A, 2 B, 3 A.

මතක තබාගන්න: Tables diagram එක නැවත අඳින්න.

Quick Quiz 03 – Security

Project එක local-only ලෙස පාලනය කිරීම වැදගත්.

දැනගත යුතු කරුණු

- 1. No login ERP එක public publish කරන්න (A) ඔව් (B) නැහැ.
- 2. SQL parameter placeholders වටිනවා (A) ඔව් (B) නැහැ.
- 3. Unverified AI action allow කරන්න (A) ඔව් (B) නැහැ.



✎ **ඔබේ අභ්‍යාසය**

Answer key: 1 B, 2 A, 3 B.

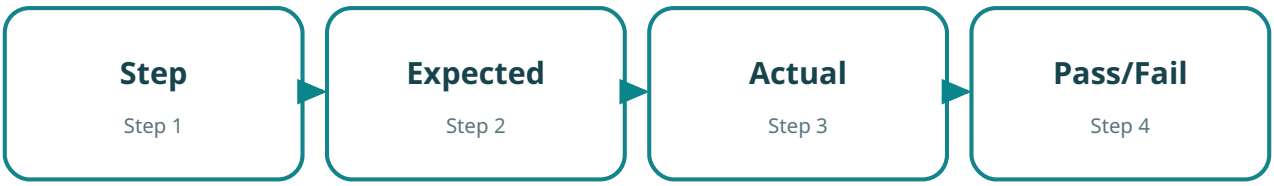
මතක තබාගන්න: Security එකේ goal එක තේරුම් ගන්න.

Quick Quiz 04 – Testing

QA role එකක් සිතමු.

දැනගත යුතු කරුණු

- 1. Invalid stock test = (A) negative test (B) logo test.
- 2. Code change පසු old tests = (A) regression (B) decoration.
- 3. Expected vs actual mismatch = (A) bug candidate (B) always correct.



✎ ඔබේ අභ්‍යාසය

Answer key: 1 A, 2 A, 3 A.

මතක තබාගන්න: TC-01 එකකට expected result එකක් ලියන්න.

Hand Math Challenge 01

Blue Pen: start 0, buy 12, sell 4, buy 3, sell 5.

දැනගත යුතු කරුණු

- Stock balance = 0 + 12 - 4 + 3 - 5 = 6.
- Selling price Rs 80 නම් total sold 9 → revenue Rs 720.
- Purchase cost වෙනම මැනීම නිසා revenue=profit නොවේ.

START		PURCHASE		FINAL STOCK
0	+10	10	-2	8
pens		pens		pens

✎ ඔබේ අභ්‍යාසය

ඔබේ calculator භාවිතා කර balance සහ revenue පරීක්ෂා කරන්න.

මතක තබාගන්න: Math checks software test expectations හදනවිට වටිනවා.

Hand Math Challenge 02

Notebook: start 0; buy 8; sell 3; sell 2.

දැනගත යුතු කරුණු

- Stock = 8 - 3 - 2 = 3.
- Selling price Rs 160; sold 5 → revenue Rs 800.
- Reorder level 4 නම් stock 3 නිසා low-stock alert.



✎ ඔබේ අභ්‍යාසය

Numbers source project එකට add කර browser report සමඟ සසඳන්න.

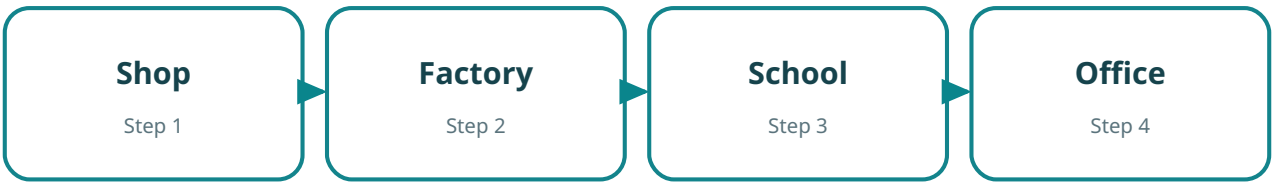
මතක තබාගන්න: Expected output කලින් අතින් සොයා තිබීම QA සඳහා ලොකු වාසියකි.

Mini Project A – School Canteen ERP

දැන් sample shop idea එක school canteen එකකට පරිවර්තනය කර බලන්න.

දැනගත යුතු කරුණු

- Products: sandwich, juice, bun.
- Purchasing: ingredients/inventory simplified.
- Sales: quantity, revenue; Reports: low-stock items.



✎ ඔබේ අභ්‍යාසය

SRS functional requirements 5ක් සහ screen sketch 1ක් හදන්න.

මතක තබාගන්න: Food expiry සහ food safety real production හි අමතර rules ඉල්ලයි.

Mini Project B – Library Supplies

Library supplies සඳහාත් shared database ideas භාවිතා කරන්න.

දැනගත යුතු කරුණු

- Products: notebooks, markers, printer paper.
- Receive = add stock; issue = reduce stock.
- Reports = available supplies සහ reorder alerts.



✎ ඔබේ අභ්‍යාසය

“Sale” කියන වචනය “Issue” ලෙස වෙනස් කරන requirements තුනක් ලියන්න.

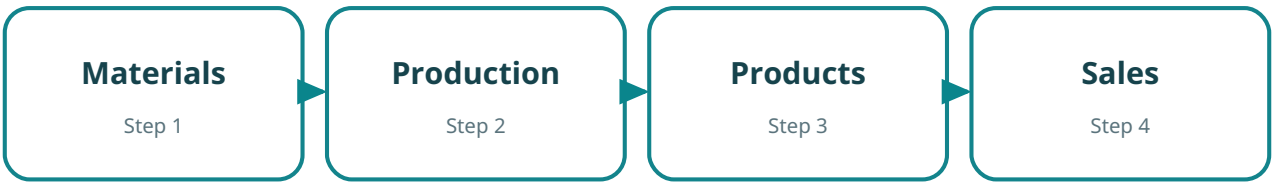
මතක තබාගන්න: පළමුව business process දැනගන්න, පසුව UI වෙනස් කරන්න.

Mini Project C – Small Bakery

Bakery එකක stock සහ sales management ගැන project plan එකක්.

දැනගත යුතු කරුණු

- Inventory: finished buns/bread units.
- Purchasing: ingredients receiving; manufacturing later.
- Sales and reports: daily quantity, revenue, wastage requirements.



✎ **ඔබේ අභ්‍යාසය**

Wastage stock movement reason එකක් ලෙස future feature එකකට යෝජනා කරන්න.

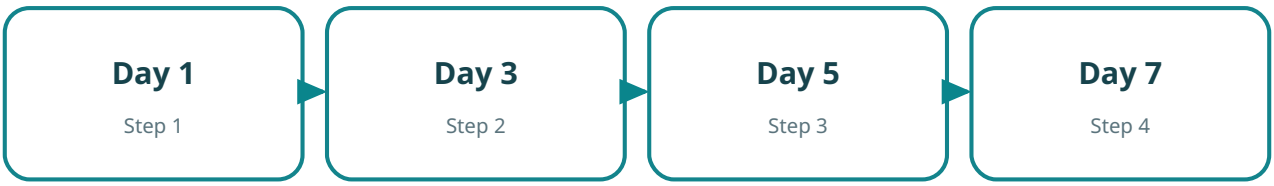
මතක තබාගන්න: Bakery manufacturing සහ perishable stock tracking මේ demo scope එකේ නැහැ.

Day 1–7 Training Plan

කුඩා step එකක් දිනපතා practice කරන්න.

දැනගත යුතු කරුණු

- Day 1 ERP concepts; Day 2 SDLC/SRS; Day 3 Python + database.
- Day 4 Products/Purchases; Day 5 Sales/Reports.
- Day 6 Testing/QA; Day 7 explain project to friend.



✎ ඔබේ අභ්‍යාසය

දින හතකට වෙන් කළ පොඩි checklist එකක් හදන්න.

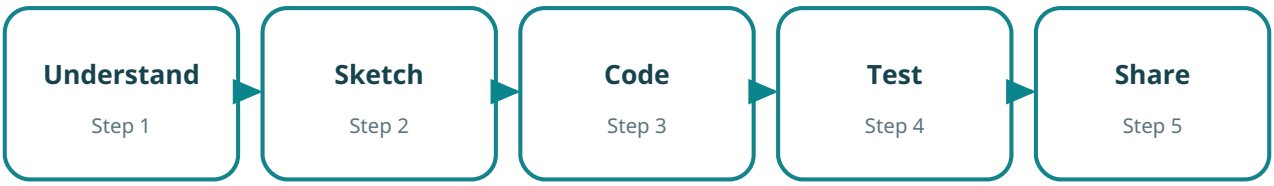
මතක තබාගන්න: ඉගෙනගත්ත දේ කෙනෙකුට පැහැදිලි කිරීම හොඳ revision ක්‍රමයක්.

Project Presentation එකක්

Software එකක් හදලා ඉවරයි කියලා පමණක් නවතින්න එපා. අර්ථයත් පෙන්වන්න.

දැනගත යුතු කරුණු

- Problem → Goal → Modules → Data flow → Demo.
- Purchase 10 → Sale 2 → Stock 8 → Reports.
- Test results + known limitations + next steps.



✎ **ඔබේ අභ්‍යාසය**

Slides නැතිව මිනිත්තු තුනක demo talk එකක් practise කරන්න.

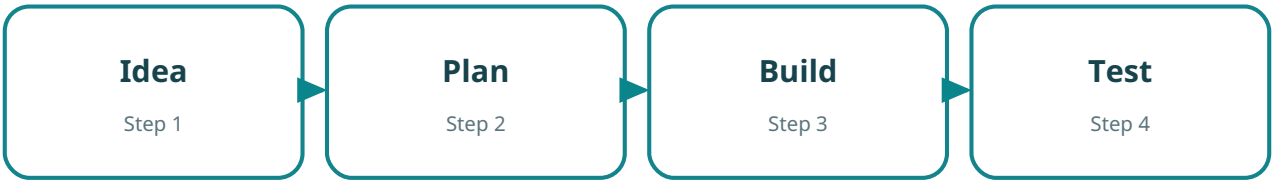
මතක තබාගන්න: Good communication කියන්නේ software engineering skill එකක්.

ඔබේ Mini ERP Certificate

ඔබ ERP concepts, local development, database, transactions සහ QA basics ඉගෙනගත්තා!

දැනගත යුතු කරුණු

- Name: _____
- Project: Little Star ERP - beginner demo.
- Skills: ERP flow · Python · SQLite · SRS · QA · Reports.



✎ ඔබේ අභ්‍යාසය

ඔබේ නම සහ අද ඉගෙනගත් වැදගත්ම lesson එක ලියන්න.

මතක තබාගන්න: ඊළඟට authentication, multi-item orders සහ professional accounting ගැන ඉගෙනගන්න පුළුවන්.

Source ZIP එකේ ඇති දේ

මේ PDF එකේ අත්‍යවශ්‍ය code උදාහරණ තිබෙනවා; ZIP එකේ full runnable project එක ඇත.

දැනගත යුතු කරුණු

- README.md: setup සහ limitations.
- db.py, app.py, schema.sql, assets/style.css.
- tests.py සහ docs folder එකේ SRS, QA, data dictionary.

VS Code · EXPLORER

```
EGOTECHWORLD_Little_Star_ERP_2026/
  |-- app.py
  |-- db.py
  |-- schema.sql
  |-- tests.py
  |-- run_windows.bat
  |-- assets/
  |    `-- style.css
  |-- docs/
  |    |-- SRS.md
  |    |-- QA_Test_Plan.md
  |    `-- Data_Dictionary.md
  `-- data/  (created when app runs)
```

✎ ඔබේ අභ්‍යාසය

ZIP එක extract කර files හඳුනාගෙන check කරන්න.

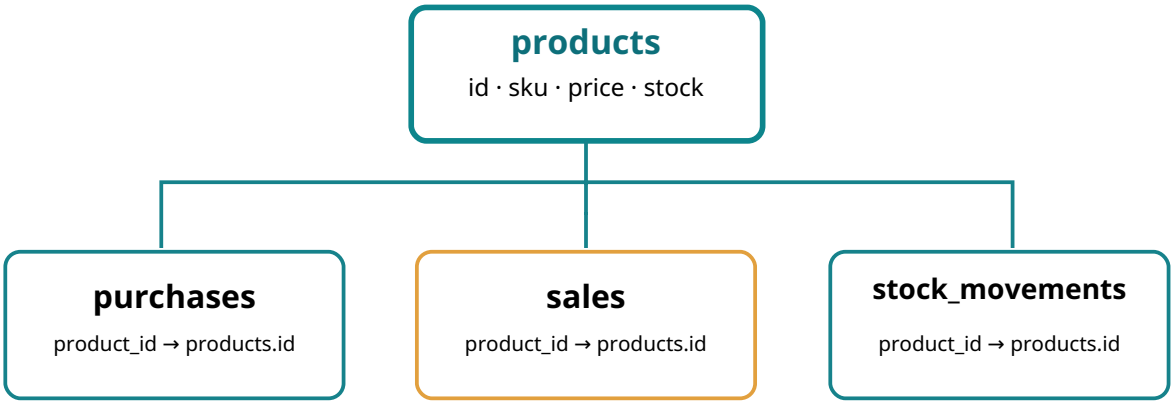
මතක තබාගන්න: Source appendix snippets copy කිරීම වෙනුවට run-ready complete files ZIP එකෙන් භාවිතා කරන්න.

SQL schema: tables map

Tables හතරේ key fields එකම page එකක සාරාංශ කරමු.

දැනගත යුතු කරුණු

- products(id,sku,name,price_cents,stock_qty,reorder_level).
- purchases(id,product_id,qty,unit_cost_cents,supplier).
- sales(id,product_id,qty,unit_price_cents,total_cents); stock_movements(product_id,change_qty,reason).



✎ ඔබේ අභ්‍යාසය

products.id සිට අනෙක් tables වෙත arrows අඳින්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Database functions map

db.py එකේ functions නමෙන්ම වැඩ හඳුනාගන්න.

දැනගත යුතු කරුණු

- `initialize` → first-run tables + products.
- `list_products`, `add_product`, `buy_stock`, `sell_stock`.
- `summary`, `recent_transactions` → dashboard and reports.

Visual Studio Code

db.py · function map

EXPLORER

Little-Star-ERP/
|-- app.py
|-- db.py
|-- schema.sql
|-- tests.py
`-- docs/

```
# Project APIs exposed by db.py
initialize()
list_products()
add_product(sku, name, price_cents, reorder_level)
buy_stock(product_id, qty, unit_cost_cents, supplier)
sell_stock(product_id, qty, customer)
summary()
recent_transactions()
```

✎ ඔබේ අභ්‍යාසය

`sell_stock` function එකේ parameters තුන නම් කරන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Browser routes map

app.py තුළ path අනුව page එකක් තෝරා HTML response ලබා දෙනවා.

දැනගත යුතු කරුණු

- `GET /` -> Dashboard; `/products` -> Product Master.
- `GET /purchases` -> Purchasing; `/sales` -> Sales.
- `GET /reports` -> Reports; POST form routes validate and save.

Visual Studio Code

app.py · route concept

```
EXPLORER

Little-Star-ERP/
|-- app.py
|-- db.py
|-- schema.sql
|-- tests.py
`-- docs/
```

```python
# The concept behind simple GET routing.
routes = {
    '/': dashboard,
    '/products': products_page,
    '/purchases': purchases_page,
    '/sales': sales_page,
    '/reports': reports_page,
}
```

> ✎ **ඔබේ අභ්‍යාසය**
>
> Browser එකේ address bar එකේ `/reports` path එක open කරන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Stock update SQL

Purchase සහ Sale දෙකම same current stock field එක වෙනස් කරනවා.

දැනගත යුතු කරුණු

- Purchase: stock_qty + quantity.
- Sale: stock_qty - quantity.
- Transactions තුළ current stock check කර update කරන්න.

Visual Studio Code

stock_updates.sql

EXPLORER

Little-Star-ERP/
|-- app.py
|-- db.py
|-- schema.sql
|-- tests.py
`-- docs/

```sql
-- After receiving stock:
UPDATE products
SET stock_qty = stock_qty + ? WHERE id = ?;

-- After a validated sale:
UPDATE products
SET stock_qty = stock_qty - ? WHERE id = ?;
```

> ✎ **ඔබේ අභ්‍යාසය**
>
> මෙම SQL දෙක වෙළඳසැලේ scenario එකට සම්බන්ධ කරන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Server start script

run_windows.bat එක තුළ command එකක් පමණයි.

දැනගත යුතු කරුණු

- `.bat` file එක double-click කිරීමෙන් local server run කළ හැක.
- CMD window එක open තබා browser open කරන්න.
- දවස් අවසානයේ Ctrl+C.

```
Windows Command Prompt                                    _ □ ×

C:\Users\Student> @echo off
C:\Users\Student> REM Start the local ERP app on Windows.
C:\Users\Student> py app.py
C:\Users\Student> pause
```

> ✎ **ඔබේ අභ්‍යාසය**
>
> File එක text editor එකෙන් open කර commands කියවන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Test command + expected run

Automated tests තුළ purchase/sale, invalid price, duplicate SKU, HTML escaping වැනි checks ඇත.

දැනගත යුතු කරුණු

- Command: py -m unittest tests.py -v.
- Automated unit/database checks = 18 in the bundled version.
- Local Windows manual test results environment අනුව වෙනස් විය හැක.

```
Windows Command Prompt                                    _ □ ×

C:\Users\Student> py -m unittest tests.py -v
```

> ✎ **ඔබේ අභ්‍යාසය**
>
> Tests pass නම් screenshot එකක් ගන්න; fail නම් error reason එක ලියන්න.

මතක තබාගන්න: පළමුව අනුමාන කරන්න. පසුව example project එකෙන් පරීක්ෂා කරන්න.

Important limitations

Mini ERP එක සම්පූර්ණ ERP එකක් ලෙස වැරදිව භාවිතා නොකළ යුතුයි.

දැනගත යුතු කරුණු

- No login or role permissions: do not use publicly.
- No payment, invoicing, taxation, VAT or general ledger.
- No backups automation, no accounting-compliant inventory valuation.



✎ **ඔබේ අභ්‍යාසය**

Production release එකකට අවශ්‍ය missing features 5ක් ලියන්න.

මතක තබාගන්න: මේ project එකෙන් data integration concept එක ඉගෙනගැනීම ප්‍රධාන අරමුණයි.

2026 ඉගෙනුම් මූලාශ්‍ර

ERP පිළිබඳ ඉදිරියට කියවීමට සහ තාක්ෂණික documentation සඳහා මුල් sources බලන්න.

දැනගත යුතු කරුණු

- SAP: https://www.sap.com/products/erp/what-is-erp.html
- Microsoft Dynamics: https://www.microsoft.com/en-us/dynamics-365/resources/what-is-erp
- Python docs: https://docs.python.org/3/library/sqlite3.html



✎ ඔබේ අභ්‍යාසය

ERP definition එකට විශ්වාසනීය source දෙකක් සසඳන්න.

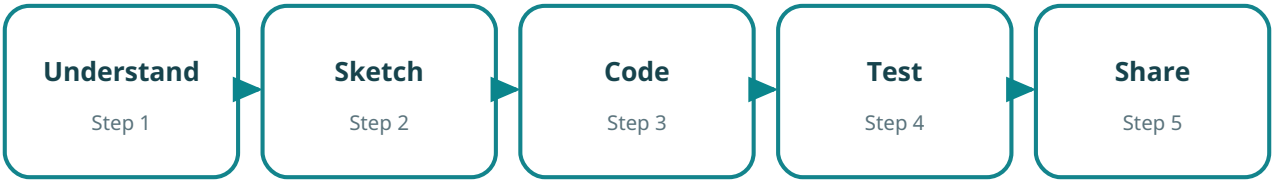
මතක තබාගන්න: මෙම පොත 2026 beginner learning material එකක්; supplier product features වෙනස් විය හැක.

ඊළඟ ඉගෙනීම

ERP Basics ඉගෙනගත් පසු features එකින් එක වැඩි කරමින් software engineering practice කරන්න.

දැනගත යුතු කරුණු

- Level 2: users/login සහ role permission system.
- Level 3: multi-item orders, return/refund, backups සහ audit.
- Level 4: professional accounting, integrations, deployment and compliance.



✎ **ඔබේ අභ්‍යාසය**

ඔබේ dream ERP එකේ ඉදිරි feature තුනක් නම් කරන්න.

මතක තබාගන්න: Start small, test often, document clearly.