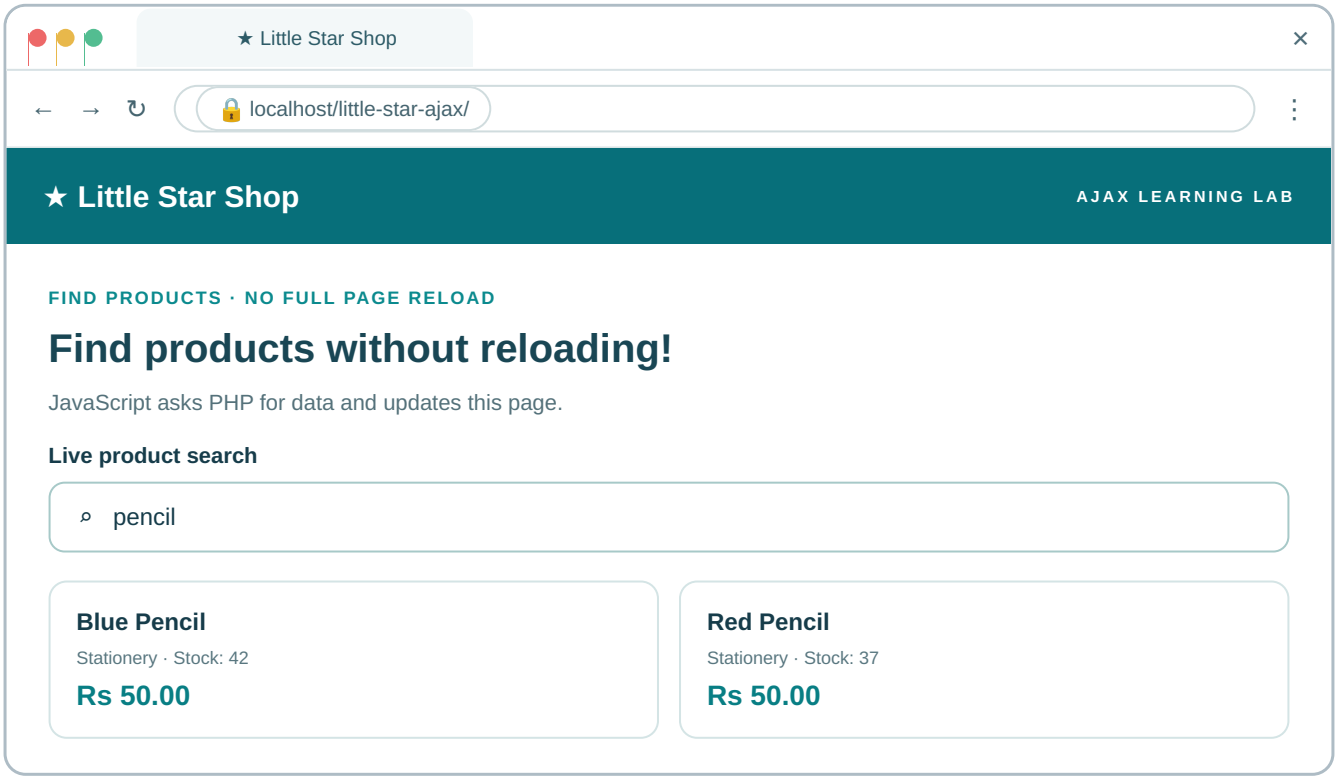
EGOTECHWORLD
2026 LEARNING SERIES

BEGINNER WORKBOOK · SINHALA

AJAX
මූල සිටම

JavaScript → fetch() → JSON → PHP → MySQL

සරල button එකකින් පටන්ගෙන, page reload නොවී products සොයන සැබෑ mini-project එකක් දක්වා.



ඉගෙනීම සඳහා සැකසූ browser preview එකකි — live bank/store system එකක් නොවේ.

01 Learn the idea

02 Write the code

03 Check the result

සිංහලෙන් පැහැදිලි පියවර · White pages · Readable code · Practical exercises

BEFORE YOU START

මේ පොත භාවිතා කරන්නේ කොහොමද?

හොඳ developer කෙනෙක් වෙන්නේ code පිටපත් කිරීමෙන් පමණක් නොවේ. එක් එක් පාඩම කියවා, code ධාවනය කරලා, ප්‍රතිඵලය තේරුම් ගන්න.

01 Read → 02 Build → 03 Explain

01
කියවන්න

අලුත් වචන සිංහලෙන් තේරුම් ගන්න. Browser, Server, JSON ගැන ප්‍රශ්න අහන්න.

02
කේතය ටයිප් කරන්න

VS Code එකේ code type කරලා Save කරන්න. අලුත් file නම නිවැරදිව තබන්න.

03
Browser එකේ බලන්න

Chrome F12 → Console / Network හරහා request සහ response පරීක්ෂා කරන්න.

04
වෙනස් කරන්න

Search word, button text, JSON value මාරු කරලා නැවත run කරන්න.

අවශ්‍ය software:
Windows (හෝ Linux/macOS), VS Code, Chrome සහ XAMPP. සරල මුල් HTML පාඩම් සඳහා XAMPP අනිවාර්ය නැහැ.

වැදගත්:
ඔබේ source ZIP හි file names මේ පොතේ උදාහරණ සමඟ ගැළපේ. Sample data පමණක් භාවිතා කරන්න.

12 LEARNING MODULES

ඔබේ AJAX ඉගෙනුම් මාර්ග සිතියම

මුල් පාඩම් browser එකේ basics; මැද පාඩම් HTTP + PHP/MySQL; අවසානයේ ක්‍රියාත්මක project එක සහ QA.

01 Web එකේ මූලික අදහස්
Browser සහ Server

02 HTML, CSS සහ JavaScript
Page එකේ අංග

03 AJAX හඳුනාගනිමු
No page reload

04 fetch() සහ async/await
Modern JavaScript

05 JSON සහ HTTP
Data format

06 PHP backend මූල සිට
Server side

07 MySQL සහ Database
Data layer

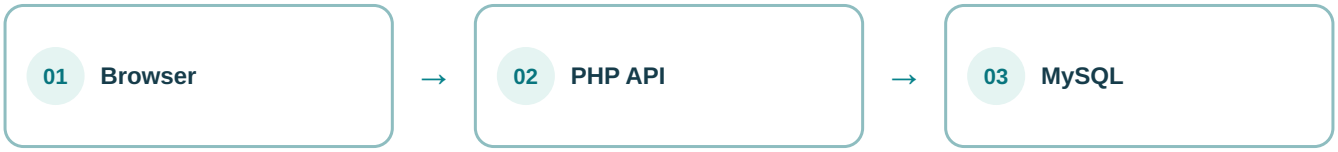
08 Little Star Shop ගොඩනගමු
Project files

09 Live Search හදමු
Search experience

10 POST form සහ JSON API
Send data

11 Security සහ Privacy
Safe beginners

12 Testing, Git සහ Next Steps
Finish well



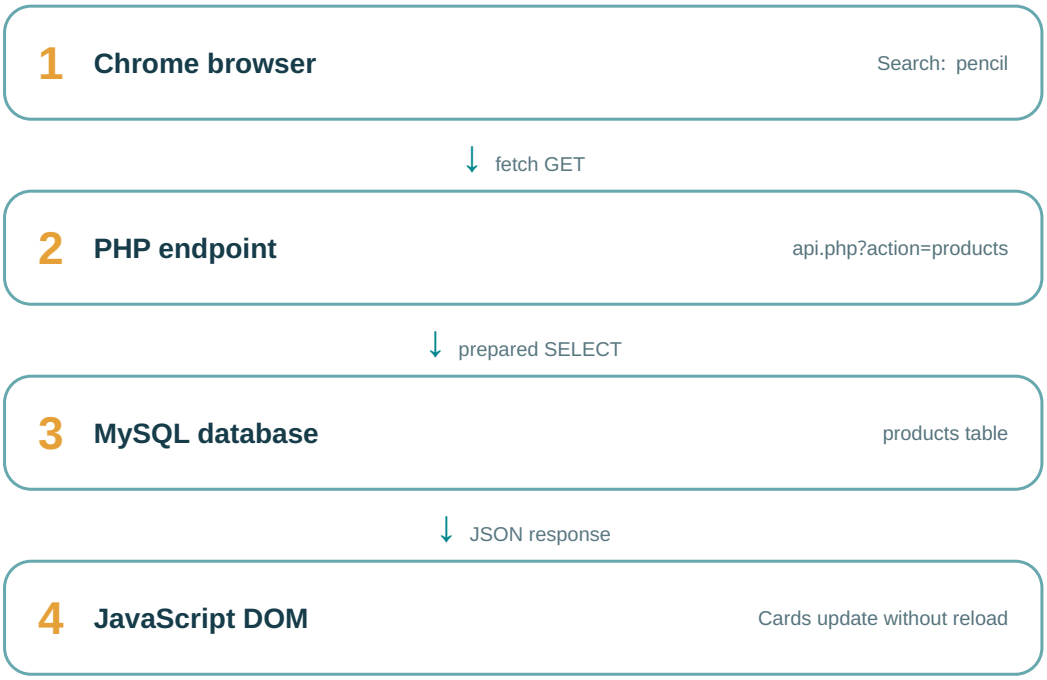
පොතේ අරමුණ

“AJAX” කියන්නේ වෙනම භාෂාවක් නොව, JavaScript මඟින් server එක සමඟ asynchronous data exchange කරන ආකාරයක් බව ප්‍රායෝගිකව තේරුම් ගැනීමයි.

THE BIG PICTURE

AJAX Request එක ගමන් කරන්නේ කොහොමද?

ඔබ “pencil” ලියන විට page එක reload නොවී product cards වෙනස් වීමට පසුපස සිදුවන දේ බලමු.



ඔබ මතක තබාගන්න:

Browser එක MySQL වෙත සෘජුව සම්බන්ධ නොවෙයි. Database එකට queries යවන්නේ backend PHP එකයි. Browser එකට JSON data පමණක් ලැබේ.

TOOLS & FILES

VS Code, Chrome සහ XAMPP සූදානම් කරමු

1 · VS Code

Project folder එක Open Folder කරන්න.

`index.php` සහ `assets/app.js` files බලන්න.

2 · XAMPP

Control Panel විවෘත කර **Apache** හා **MySQL** start කරන්න.

3 · Project copy

ZIP unzip කර folder එක `C:\xampp\htdocs\little-star-ajax` යට තබන්න.

4 · Database

phpMyAdmin → Import → `sql/database.sql`.

Local `config.php` සකස් කරන්න.

EXPLORER / PROJECT — folder-tree.txt

```
little-star-ajax/
├── index.php          # Main web page
├── api.php            # JSON API
├── db.php             # Database connection
├── helpers.php        # Validation helpers
├── assets/
│   ├── app.js         # AJAX logic
│   └── style.css      # Interface
├── lessons/           # First practice files
└── sql/database.sql   # Database tables
```

UTF-8 • folder-tree.txt • EGOTECHWORLD LEARNING LAB

Open URL:

`http://localhost/little-star-ajax/` — `file://` මඟින් API දත්ත load වන්නේ නැහැ.

MODULE 01 / 12

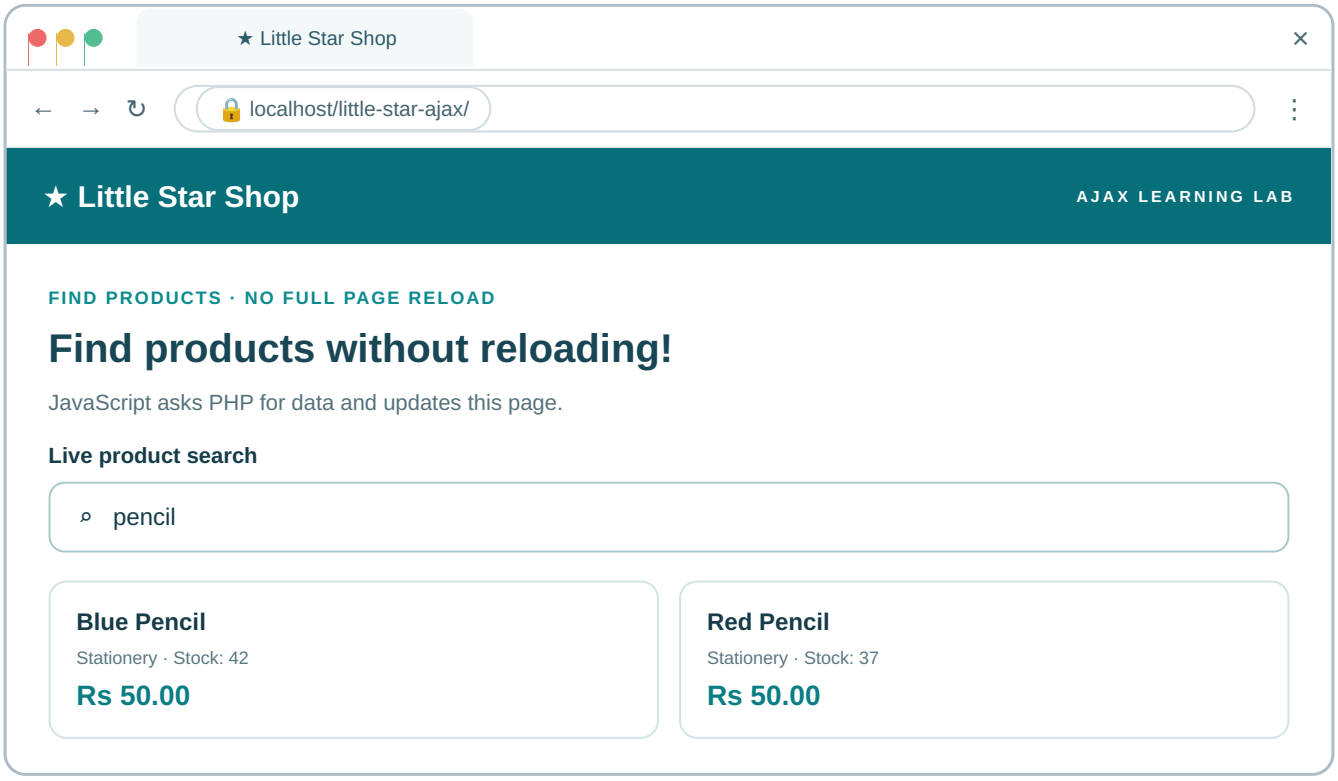
YOUR NEXT STEP

Browser සහ Server

01 Web එකේ මූලික අදහස්

Web page එකක් අපේ පරිගණකයට ලැබෙන ගමන තේරුම් ගනිමු.

01 Browser → 02 HTTP request → 03 Server



ඉගෙනීම සඳහා සැකසූ browser preview එකකි — live bank/store system එකක් නොවේ.

★ THIS MODULE'S MISSION

Chrome F12 විවෘත කර Console එක බලන්න.

01.01 Website එකක් යනු කුමක්ද?

Website එක HTML, CSS සහ JavaScript ගොනු වලින් සෑදේ. Chrome ඒවා කියවා පිටුවක් පෙන්වයි. AJAX ඉගෙනීමට පෙර browser එකේ වැඩේ දැනගත යුතුයි.

EXPLORER / PROJECT index.html

```
1 <h1>Hello website</h1>
2 <p>Welcome to Little Star!</p>
```

UTF-8 • index.html • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Chrome විවෘත කර සාමාන්‍ය website එකක address bar එක බලන්න.

CHECK · ප්‍රතිඵලය

Address එක URL එකක් බව හඳුනාගන්න.

TIP Website එකක් සහ browser එක එකම දෙයක් නොවේ.

01.02 Browser එකේ කාර්යය

Browser එක user ගේ click, typing සහ request පාලනය කරයි. JavaScript බොහෝ විට browser තුළම ධාවනය වෙයි.

EXPLORER / PROJECT index.html

```
1 // Press F12 > Console in Chrome.
2 console.log("Hello from Chrome!");
```

UTF-8 • index.html • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Chrome එකේ F12 ඔබලා Console tab එක විවෘත කරන්න.

CHECK · ප්‍රතිඵලය

Developer Tools දැකිය හැකි වේ.

TIP Console එකේ නොදන්නා code paste කරන්න එපා.

01.03 Server එක යනු කුමක්ද?

Server එක browser ඉල්ලන දත්ත හෝ HTML ලබා දෙයි. අපේ PHP lesson වල XAMPP Apache server එක ඒ වැඩේ කරයි.

EXPLORER / PROJECT — index.html

```
Browser  ->  GET /index.php
Server   ->  HTTP 200 + HTML
Browser  ->  Paint page
```

UTF-8 • index.html • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

කඩදාසියක browser සහ server ලෙස කොටු දෙකක් අඳින්න.

CHECK · ප්‍රතිඵලය

ඉල්ලීම සහ පිළිතුර අතර ඊතල දෙකක් ඇඳේ.

TIP Server එක Internet එකේ පමණක් නොව localhost එකේත් ධාවනය වෙයි.

01.04 URL එක කියවමු

URL එකේ protocol, host, path සහ query යන කොටස් තිබෙනවා. උදාහරණයට localhost යනු ඔබේ පරිගණකයයි.

Command Prompt — _ □ ×

```
C:\Users\Student> http://localhost/little-star-ajax/
       host        path
```

DO IT · ඔබ කරන්න

http://localhost/little-star-ajax/ ලෙස browser එකේ ටයිප් කරන්න.

CHECK · ප්‍රතිඵලය

Project එක install කර ඇති විට app එක විවෘත වේ.

TIP file:// හා http:// එක සමාන නොවේ.

01.05 HTTP request එක

Request යනු browser එක server එකෙන් data ඉල්ලන පණිවිඩය. GET කියන්නේ කියවීම සඳහා බහුල method එකයි.

EXPLORER / PROJECT index.html

```
GET /api.php?action=products HTTP/1.1
Host: localhost
```

UTF-8 • index.html • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

කඩදාසියේ GET /products ලෙස ලියන්න.

CHECK · ප්‍රතිඵලය

Request එකේ path සහ method වෙන් කර හඳුනාගන්න.

TIP GET මඟින් රහස් තොරතුරු යවන්න එපා.

01.06 HTTP response එක

Server එක status code එකක් සහ body එකක් ලබා දෙයි. 200 සාර්ථකයි; 404 මාර්ගය නොමැත; 500 server ගැටලුවකි.

EXPLORER / PROJECT index.html

```
HTTP/1.1 200 OK
Content-Type: application/json

{"ok":true}
```

UTF-8 • index.html • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

200, 404, 500 අංක වලට අරුත් ලියන්න.

CHECK · ප්‍රතිඵලය

මූලික status 3 හඳුනාගන්න පුළුවන්.

TIP හැම error එකක්ම JavaScript bug එකක් නොවේ.

01.07 Page reload යනු?

සාමාන්‍ය link එකක් අලුත් page එකක් load කරයි. AJAX නම් page එකේ කොටසකට අලුත් දත්ත යාවත්කාලීන කළ හැකියි.

EXPLORER / PROJECT index.html

```
Traditional: submit -> full page reload
AJAX:        fetch -> update one section
```

UTF-8 • index.html • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Search box එකට type කළ විට page reload අවශ්‍ය දැයි සිතන්න.

CHECK · ප්‍රතිඵලය

AJAX යොදාගන්නා අවස්ථා හඳුනාගන්න.

TIP AJAX යනු වෙනම programming language එකක් නොවේ.

★ MODULE 01 CHECKPOINT

මෙම module එක අවසන්! පහත ප්‍රශ්නවලට පිළිතුරු ලියා බලා ඉදිරියට යන්න.

1. Browser එකේ කාර්යය ගැන උදාහරණයක් දෙන්න.
2. URL එක කියවමු ඔබේම වචනවලින් පැහැදිලි කරන්න.
3. **Mini task:** Chrome F12 විවෘත කර Console එක බලන්න.

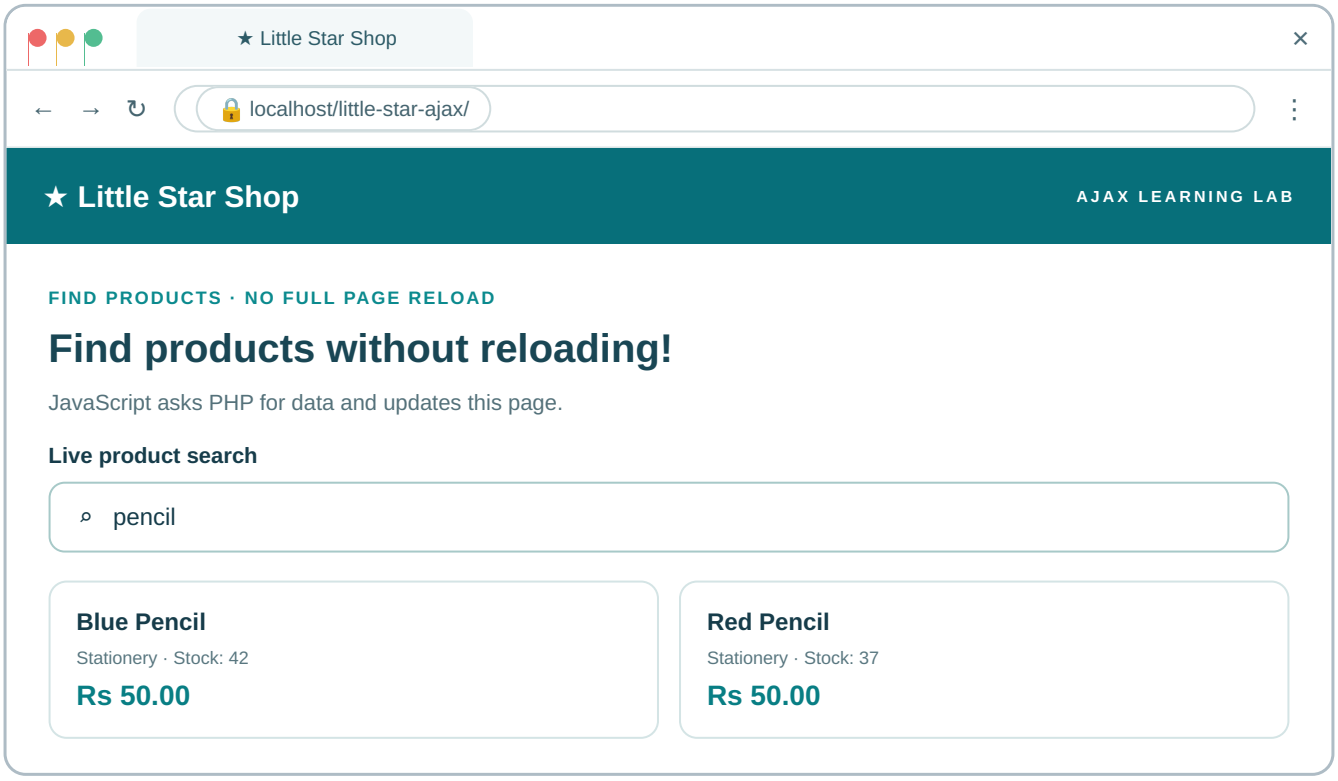
☐ Read ☐ Run ☐ Explain ☐ Change one line

Page එකේ අංග

02 HTML, CSS සහ JavaScript

AJAX කරන්න කලින් බොත්තම, input සහ result box එක සකස් කරමු.

01 HTML → 02 JavaScript → 03 Result



ඉගෙනීම සඳහා සැකසූ browser preview එකකි — live bank/store system එකක් නොවේ.

★ THIS MODULE'S MISSION

Button click කළ විට text වෙනස් වෙන file එක සාදන්න.

02.01 HTML වලින් ආරම්භය

HTML යනු page එකේ ව්‍යුහයයි. h1 heading, p paragraph, button click කළ හැකි control එකක්.

EXPLORER / PROJECT index.html

```html
<h1>Little Star</h1>
<button id="hello">Click</button>
<p id="message">Waiting...</p>
```

UTF-8 • index.html • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

lessons/01_hello_ajax.html file එක browser එකෙන් විවෘත කරන්න.

CHECK · ප්‍රතිඵලය

Button සහ Waiting... පණිවිඩය පෙනේ.

TIP HTML tag එකක් වසා දැමීමට අමතක කරන්න එපා.

02.02 CSS වලට කාර්යයක් ඇත

CSS අකුරු, වර්ණ, ඉඩකඩ සහ responsiveness පාලනය කරයි. CSS එක AJAX data ගෙන එන්නේ නැහැ.

EXPLORER / PROJECT app.js

```css
/* Simple, readable page style. */
h1 { color: #087783; }
button { padding: 12px; }
```

UTF-8 • app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

h1 එකට color property එකක් අලුතින් එකතු කරන්න.

CHECK · ප්‍රතිඵලය

Heading එකේ වර්ණය වෙනස් වේ.

TIP Style වෙනස් කළා කියලා server data වෙනස් වෙන්නේ නැහැ.

02.03 JavaScript එක මොකක්ද?

JavaScript click එකකට ප්‍රතිචාර දක්වයි. Data ලබාගැනීමට fetch() යොදාගන්නත් JavaScript භාවිතා කරනවා.

EXPLORER / PROJECT app.js

```
1  // JavaScript runs in the browser.
2  console.log("Hello!");
```

UTF-8 • app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Chrome Console එකේ console.log("Hello") ලියන්න.

CHECK · ප්‍රතිඵලය

Hello යන්න Console එකට මුද්‍රණය වේ.

TIP console.log යනු user interface එක නොව developer message එකකි.

02.04 Variable එකක්

let/const භාවිතයෙන් value එකකට නමක් දෙනවා. const අගය නැවත assign නොකරන තැන්වල හොඳයි.

EXPLORER / PROJECT app.js

```
1  const shopName = "Little Star";
2  let items = 3;
3  console.log(shopName, items);
```

UTF-8 • app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

const shop = "Little Star"; ලියන්න.

CHECK · ප්‍රතිඵලය

shop variable එකේ නම තබාගත හැකි වේ.

TIP const object එකේ සියලුම data වෙනස් කළ නොහැකි බව නොඅදහන්න.

02.05 DOM යනු?

DOM යනු HTML page එක JavaScript වලින් තෝරාගෙන වෙනස් කළ හැකි ව්‍යුහයයි. id එකකින් element එකක් පහසුවෙන් තෝරාගත හැක.

EXPLORER / PROJECT app.js

```
1  const message = document.querySelector("#message");
2  message.textContent = "Hello!";
```

UTF-8 • app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

document.querySelector("#message") ප්‍රතිඵලය බලන්න.

CHECK · ප්‍රතිඵලය

Message paragraph එක හඳුනාගනී.

TIP id වැරදි නම් null ලබාදිය හැක.

02.06 Button click event

addEventListener භාවිතයෙන් user button එකක් click කළ විට function එකක් ධාවනය කරනවා.

EXPLORER / PROJECT app.js

```
1  document.querySelector("#hello").addEventListener(
2    "click", () => {
3      document.querySelector("#message")
4        .textContent = "Hello, friend!";
5    }
6  );
```

UTF-8 • app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Hello example එකේ button click කරන්න.

CHECK · ප්‍රතිඵලය

Waiting... වෙනුවට Hello පෙනේ.

TIP Function එක වහාම නොව event එක සිදු වූ විට ධාවනය කරන්න.

02.07 textContent ආරක්ෂාව

Server හෝ user text එක page එකට පෙන්වන විට textContent භාවිතා කළොත් HTML code ලෙස ක්‍රියා නොකරයි.

EXPLORER / PROJECT app.js

```
// Safer than innerHTML for untrusted values.
message.textContent = "<b>Hello</b>";
```

UTF-8 • app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Message එකට Hello text එක ලබා දෙන්න.

CHECK · ප්‍රතිඵලය

Tag එක text ලෙසම පෙන්වයි.

TIP අවිශ්වාස input innerHTML තුළ නොදමන්න.

★ MODULE 02 CHECKPOINT

මෙම module එක අවසන්! පහත ප්‍රශ්නවලට පිළිතුරු ලියා බලා ඉදිරියට යන්න.

1. CSS වලට කාර්යයක් ඇත ගැන උදාහරණයක් දෙන්න.
2. Variable එකක් ඔබේම වචනවලින් පැහැදිලි කරන්න.
3. **Mini task:** Button click කළ විට text වෙනස් වෙන file එක සාදන්න.

☐ Read ☐ Run ☐ Explain ☐ Change one line

MODULE 03 / 12

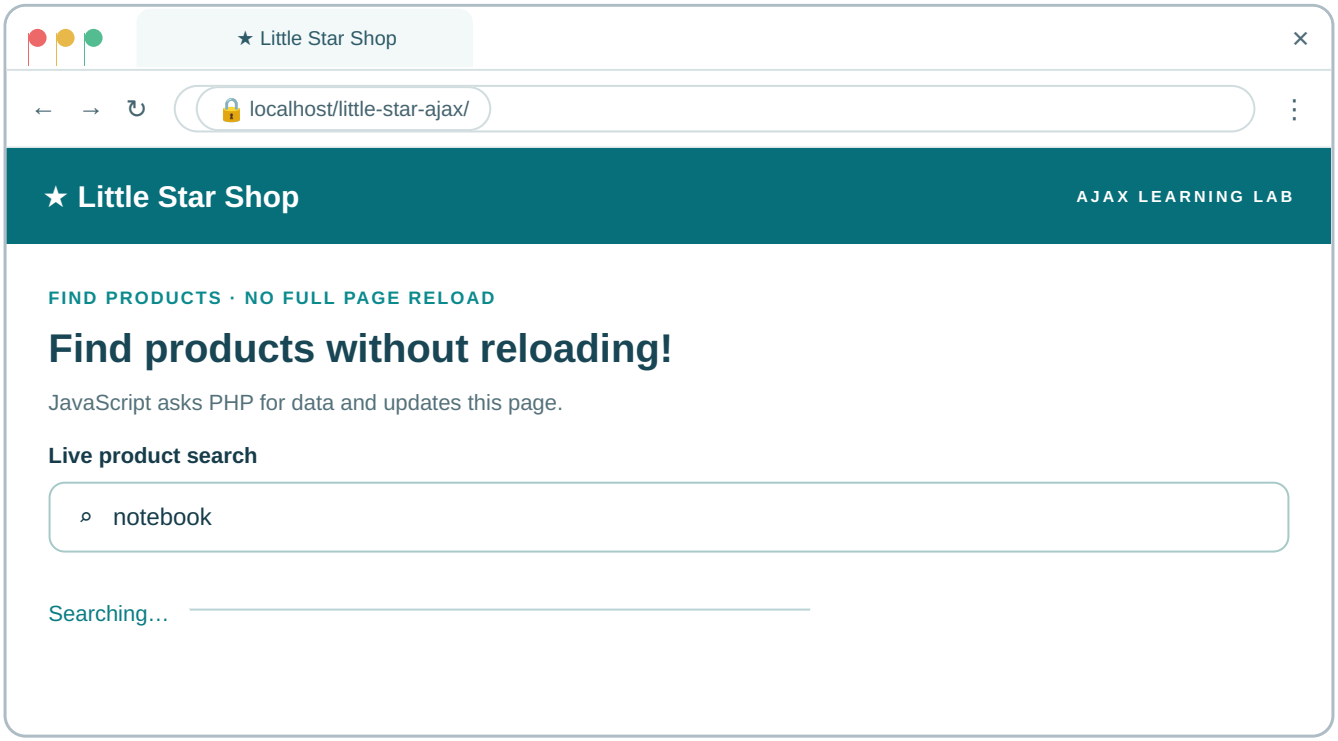
YOUR NEXT STEP

No page reload

03 AJAX හඳුනාගනිමු

දත්ත නැවත පෙන්වීම සඳහා මුළු පිටුවම reload නොකරන ක්‍රමය.

01 User event → 02 AJAX request → 03 DOM update



ඉගෙනීම සඳහා සැකසූ browser preview එකකි — live bank/store system එකක් නොවේ.

★ THIS MODULE'S MISSION

Normal reload සහ AJAX update අතර වෙනස අඳින්න.

03.01 AJAX කියන්නේ මොකක්ද?

AJAX යනු Asynchronous JavaScript and XML යන පැරණි නාමයයි. අද AJAX වල JSON බහුලව භාවිතා වන අතර JavaScript fetch() බොහෝ විට යොදාගන්නවා.

EXPLORER / PROJECT — notes.txt

```
1  // 2026 AJAX: no full page reload needed.
2  fetch("sample.json").then(r => r.json());
```

UTF-8 • notes.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

AJAX නාමයේ අකුරු සහ එහි අද අදහස කියන්න.

CHECK · ප්‍රතිඵලය

පිටුව reload නොකර දත්ත ලබාගැනීම බව තේරේ.

TIP AJAX භාවිතයට XML අනිවාර්ය නොවේ.

03.02 AJAX අවශ්‍ය ඇයි?

Search suggestions, notification counters, live filters සහ comment form වැනි තැන්වල අලුත් data පමණක් ලබාගැනීම පහසුයි.

EXPLORER / PROJECT — notes.txt

```
1  // Live search, counters, suggestion forms.
2  // All may use AJAX + JSON.
```

UTF-8 • notes.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

ඔබ දකින apps තුනක AJAX භාවිතයක් සිතන්න.

CHECK · ප්‍රතිඵලය

Search, chat, dashboard උදාහරණ හඳුනාගනී.

TIP සෑම කාර්යයකටම AJAX අවශ්‍ය නැහැ.

03.03 Asynchronous යනු?

Request එක යවන විට browser එක වෙනත් වැඩ නවත්වන්නේ නැහැ. පිළිතුර ලැබෙන තුරු program එක වෙනත් user actions පිළිගනී.

EXPLORER / PROJECT notes.txt

```
1  // While waiting, the user can keep typing.
2  const responsePromise = fetch("sample.json");
```

UTF-8 • notes.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Request අතරතුර typing කළ හැකි බව පැහැදිලි කරන්න.

CHECK · ප්‍රතිඵලය

Waiting time එක UI block නොකරන බව තේරේ.

TIP Asynchronous කියන්නේ response එක සෑමවිටම වේගවත් බව නොවේ.

03.04 Traditional form vs AJAX

සාමාන්‍ය form submit එකක browser navigation සිදු විය හැකියි. AJAX submit එකක් JavaScript වලින් message පමණක් පෙන්විය හැකියි.

EXPLORER / PROJECT notes.txt

```
1  // Traditional: browser moves to another page.
2  // AJAX: browser reads data and updates DOM.
```

UTF-8 • notes.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

වගුවක traditional හා AJAX ක්‍රම දෙක සසඳන්න.

CHECK · ප්‍රතිඵලය

Page reload වෙනස පැහැදිලි වේ.

TIP Form validation දෙපසම අවශ්‍යයි.

03.05 Request journey

User typing → JavaScript → PHP API → MySQL → JSON → JavaScript → screen ලෙස දත්ත ගමන් කරයි.

EXPLORER / PROJECT notes.txt

```
Input -> JavaScript -> PHP -> MySQL
     <-   JSON       <-
```

UTF-8 • notes.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

ඊතල හතක් සමඟ flowchart එකක් අඳින්න.

CHECK · ප්‍රතිඵලය

Frontend, backend සහ database වෙන් කර හඳුනාගනී.

TIP MySQL browser එකෙන් කෙළින්ම call නොකරයි.

03.06 XHR සහ fetch()

පැරණි AJAX සඳහා XMLHttpRequest (XHR) යොදාගත්තත් නව projects වල fetch() සහ async/await සාමාන්‍යයෙන් පහසුයි.

EXPLORER / PROJECT notes.txt

```
// Modern AJAX uses fetch and async/await.
const result = await fetch("sample.json");
```

UTF-8 • notes.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

XHR කියන නම හා fetch කියන නම ලියන්න.

CHECK · ප්‍රතිඵලය

නව උදාහරණ සඳහා fetch තෝරාගන්න පුළුවන්.

TIP fetch() request එයම database එකක් නොවේ.

03.07 AJAX limitations

Internet/server අක්‍රිය නම් request අසමත් විය හැකියි. Loading, error සහ retry messages user ට පෙන්වීම අත්‍යවශ්‍යයි.

EXPLORER / PROJECT notes.txt

```
// Always provide a failure message.
try { /* request */ }
catch { alert("Please retry"); }
```

UTF-8 • notes.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Wi-Fi off කළ විට app එක කෙසේ හැසිරිය යුතුද?

CHECK · ප්‍රතිඵලය

Error පෙන්විය යුතු බව තේරේ.

TIP Success message එක server confirmation නැතිව පෙන්වන්න එපා.

★ MODULE 03 CHECKPOINT

මෙම module එක අවසන්! පහත ප්‍රශ්නවලට පිළිතුරු ලියා බලා ඉදිරියට යන්න.

1. AJAX අවශ්‍ය ඇයි? ගැන උදාහරණයක් දෙන්න.
2. Traditional form vs AJAX ඔබේම වචනවලින් පැහැදිලි කරන්න.
3. **Mini task:** Normal reload සහ AJAX update අතර වෙනස අඳින්න.

☐ Read ☐ Run ☐ Explain ☐ Change one line

MODULE 04 / 12

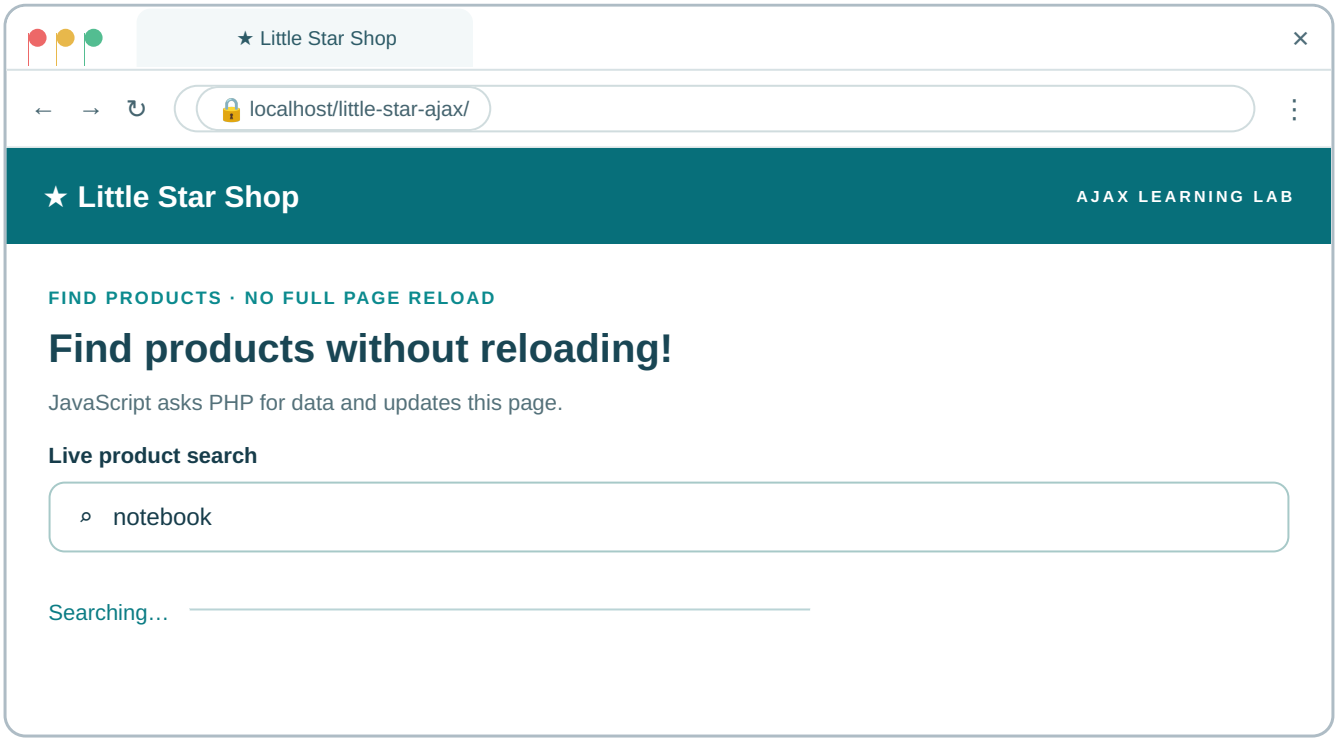
YOUR NEXT STEP

Modern JavaScript

04 fetch() සහ async/await

Browser එකේ සිට JSON file එකකට හෝ PHP endpoint එකකට request යවමු.

01 fetch() → 02 Response → 03 await json()



ඉගෙනීම සඳහා සැකසූ browser preview එකකි — live bank/store system එකක් නොවේ.

★ THIS MODULE'S MISSION

sample.json එක fetch කර result පෙන්වන්න.

04.01 fetch() පළමු පියවර

fetch("sample.json") ලෙස ලියන විට browser එක HTTP request එකක් යවයි. එය Promise එකක් ලබාදෙනවා.

EXPLORER / PROJECT app.js

```
1  // Use the sample JSON file on localhost.
2  const response = await fetch("sample.json");
```

UTF-8 • app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

lessons/02_fetch_json.html localhost හරහා විවෘත කරන්න.

CHECK · ප්‍රතිඵලය

Load items click කළ විට list එක පෙන්වයි.

TIP file:// ඔස්සේ fetch local JSON කිරීමේදී browser restrictions ඇති විය හැක.

04.02 Promise එකක්

Promise යනු අනාගතයේ ලැබෙන result එකකි. pending, fulfilled හෝ rejected තත්ත්වයන් ගැන දැනගන්න.

EXPLORER / PROJECT app.js

```
1  // Promise has states:
2  // pending -> fulfilled / rejected
3  const promise = fetch("sample.json");
```

UTF-8 • app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Promise states තුන කඩදාසියේ ලියන්න.

CHECK · ප්‍රතිඵලය

සාර්ථක/අසාර්ථක request දෙක වෙන් කරන්න පුළුවන්.

TIP Promise එක JSON data එකම නොවේ.

04.03 async function

async යන keyword එක async function එකක් හඳුන්වයි. ඒ තුළ await භාවිතයෙන් code එක කියවීමට පහසු කරගන්නවා.

EXPLORER / PROJECT app.js

```
1  async function load() {
2    const response = await fetch("sample.json");
3    return response.json();
4  }
```

UTF-8 • app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

async function load() නමින් function එකක් ලියන්න.

CHECK · ප්‍රතිඵලය

await ඇතුළත් function එකක් සෑදේ.

TIP await සාමාන්‍ය function එකකට පමණක් යොදාගන්න එපා.

04.04 await response

await fetch(url) request එකේ HTTP response ලැබෙන තුරු එම async function එක බලා සිටී; browser එක සම්පූර්ණයෙන් freeze නොවේ.

EXPLORER / PROJECT app.js

```
1  const response = await fetch("sample.json");
2  console.log(response.status);
```

UTF-8 • app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Network tab එකෙන් request duration බලන්න.

CHECK · ප්‍රතිඵලය

Request කාලය වෙනස් විය හැකි බව පෙනේ.

TIP response.ok පරීක්ෂා නොකළොත් 404 silent වැරදි විය හැක.

04.05 response.ok භාවිතය

fetch() HTTP 404/500 මත ස්වයංක්‍රීයව reject නොවිය හැක. response.ok false දැයි පරීක්ෂා කළ යුතුයි.

EXPLORER / PROJECT app.js

```
1  const response = await fetch("wrong.json");
2  if (!response.ok) {
3    throw new Error(`HTTP ${response.status}`);
4  }
```

UTF-8 • app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

වැරදි URL එකක් request කර error message එක බලන්න.

CHECK · ප්‍රතිඵලය

HTTP error අල්ලා ගත හැක.

TIP Network failure සහ HTTP error එකම දෙයක් නොවේ.

04.06 response.json()

Server JSON යැවූ විට response.json() මඟින් JS object/array එකක් ලබාගන්න පුළුවන්.

EXPLORER / PROJECT app.js

```
1  const response = await fetch("sample.json");
2  const data = await response.json();
3  console.log(data.items);
```

UTF-8 • app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

sample.json එකේ Pencil වෙනුවට Pen දාලා නැවත load කරන්න.

CHECK · ප්‍රතිඵලය

ලැයිස්තුව Pen ලෙස වෙනස් වේ.

TIP Server HTML යවන්නේ නම් JSON parse fail විය හැක.

04.07 try, catch, finally

try request එක කරන්න; catch වැරදි ප්‍රදර්ශනය කරන්න; finally loading button එක නැවත enable කරන්න.

EXPLORER / PROJECT app.js

```
1  try {
2    const response = await fetch("sample.json");
3    if (!response.ok) throw new Error("HTTP error");
4  } catch (error) {
5    console.error(error.message);
6  }
```

UTF-8 • app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Server එක නවත්වලා catch message එක බලන්න.

CHECK · ප්‍රතිඵලය

User ට හොඳ error එකක් පෙනේ.

TIP Errors නිශ්ශබ්දව සඟවන්න එපා.

★ MODULE 04 CHECKPOINT

මෙම module එක අවසන්! පහත ප්‍රශ්නවලට පිළිතුරු ලියා බලා ඉදිරියට යන්න.

1. Promise එකක් ගැන උදාහරණයක් දෙන්න.
2. await response ඔබේම වචනවලින් පැහැදිලි කරන්න.
3. **Mini task:** sample.json එක fetch කර result පෙන්වන්න.

☐ Read ☐ Run ☐ Explain ☐ Change one line

MODULE 05 / 12

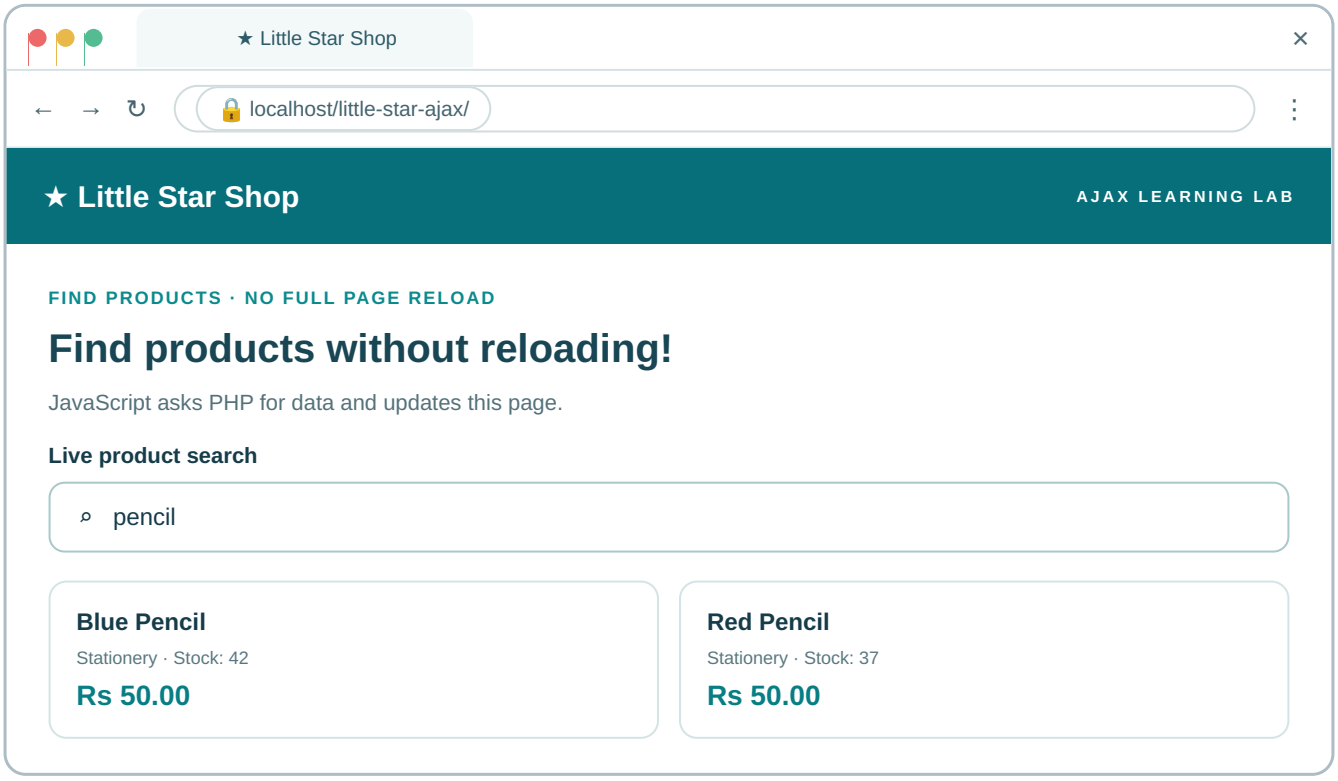
YOUR NEXT STEP

Data format

05 JSON සහ HTTP

Backend සහ frontend එකට එකම data භාෂාවක් අවශ්‍යයි.

01 PHP array → 02 JSON text → 03 JS object



ඉගෙනීම සඳහා සැකසූ browser preview එකකි — live bank/store system එකක් නොවේ.

★ THIS MODULE'S MISSION

GET/POST requests සහ JSON object එක සසඳන්න.

05.01 JSON යනු මොකක්ද?

JSON යනු text format එකකි. key:value යුගල, arrays, strings, numbers සහ booleans තබාගන්න පුළුවන්.

EXPLORER / PROJECT — sample.json

```
1 {"items":["Pencil","Notebook"]}
```

UTF-8 • sample.json • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

sample.json විවෘත කර items array එක බලන්න.

CHECK · ප්‍රතිඵලය

JSON object එකක් හඳුනාගන්න.

TIP JSON තුළ comments ඇතුළත් නොකරන්න.

05.02 JSON object එක

Object එක `{ "name": "Pencil" }` වගේ key/value සම්බන්ධය පෙන්වයි. Key එක double quotes තුළ තිබිය යුතුයි.

EXPLORER / PROJECT — sample.json

```
1 {
2   "name": "Blue Pencil",
3   "price_cents": 5000,
4   "stock_qty": 42
5 }
```

UTF-8 • sample.json • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

name සහ price සහිත JSON object එකක් ලියන්න.

CHECK · ප්‍රතිඵලය

Product data සරලව නිරූපණය වේ.

TIP JSON syntax හි trailing comma නැහැ.

05.03 JSON array එක

Array එකේ item ගණනාවක් අනුපිළිවෙළින් තබන්න පුළුවන්. Product results සාමාන්‍යයෙන් array එකක්.

EXPLORER / PROJECT sample.json

```
1 ["Pencil", "Book", "Eraser"]
```

UTF-8 • sample.json • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

පැන්සල්, පොත, මකනය යන 3ක් array එකකට දමන්න.

CHECK · ප්‍රතිඵලය

Item තුන ලැයිස්තුවක් ලැබේ.

TIP Array සහ object syntax එකම නොවේ.

05.04 JSON.stringify

JavaScript object එක POST request body එකක් ලෙස යැවීමට JSON.stringify භාවිතා කරනවා.

EXPLORER / PROJECT sample.json

```
1 const item = { name: "Pencil", price: 50 };
2 const jsonText = JSON.stringify(item);
```

UTF-8 • sample.json • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

object එකක් JSON string එකක් කරන්න.

CHECK · ප්‍රතිඵලය

"name" යන key ඇතුළත් text එක ලැබේ.

TIP JSON.stringify HTML escape ක්‍රමයක් නොවේ.

05.05 GET සහ POST වෙනස

GET සාමාන්‍යයෙන් data කියවයි. POST මඟින් අලුත් දත්ත create කිරීම වැනි server-side changes යැවිය හැක.

EXPLORER / PROJECT sample.json

```
1  // GET: read data
2  fetch("api.php?action=products");
3  // POST: create data
```

UTF-8 • sample.json • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Product search එක GETද Suggestion form එක POSTද තීරණය කරන්න.

CHECK · ප්‍රතිඵලය

GET / POST භාවිතය තේරේ.

TIP Sensitive values URL query තුළ නොදමන්න.

05.06 Status codes

200=OK, 201=Created, 400=Bad request, 403=Forbidden, 404=Not found, 422=Validation error, 500=Server error.

EXPLORER / PROJECT sample.json

```
1  200 OK          201 Created
2  403 Forbidden   404 Not Found
3  422 Validation  500 Server error
```

UTF-8 • sample.json • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

201 සහ 422 වෙනස ළඟ note එකක් ලියන්න.

CHECK · ප්‍රතිඵලය

Form submit success/error හඳුනාගනී.

TIP User-facing errors වල DB passwords පෙන්වන්න එපා.

05.07 Content-Type header

JSON යවන API එකක් `application/json; charset=utf-8` ලබාදිය යුතුයි. Browser එක data type එක නිවැරදිව තේරුම් ගනී.

EXPLORER / PROJECT api.php

```php
header("Content-Type: application/json; charset=utf-8");
echo json_encode(["ok" => true]);
```

UTF-8 • api.php • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Chrome Network > Headers තුළ Content-Type සොයන්න.

CHECK · ප්‍රතිඵලය

API response type එක පරීක්ෂා කළ හැක.

TIP HTML response එක JSON යැයි ගෑ ලෙස parse නොකරන්න.

★ MODULE 05 CHECKPOINT

මෙම module එක අවසන්! පහත ප්‍රශ්නවලට පිළිතුරු ලියා බලා ඉදිරියට යන්න.

1. JSON object එක ගැන උදාහරණයක් දෙන්න.
2. JSON.stringify ඔබේම වචනවලින් පැහැදිලි කරන්න.
3. **Mini task:** GET/POST requests සහ JSON object එක සසඳන්න.

☐ Read ☐ Run ☐ Explain ☐ Change one line

MODULE 06 / 12

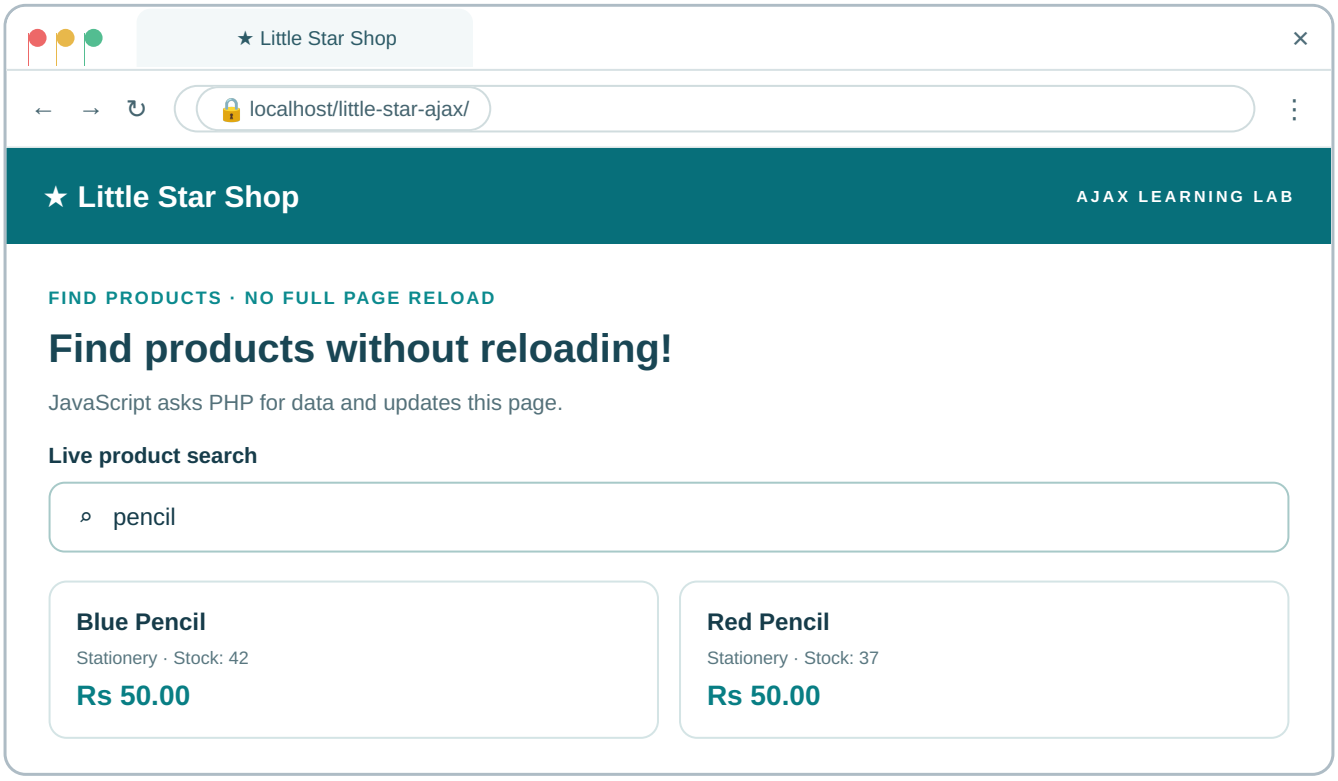
YOUR NEXT STEP

Server side

06 PHP backend මුල සිට

PHP file එකක් Apache මඟින් ධාවනය කර JSON API එකක් සාදමු.

01 Chrome → 02 Apache + PHP → 03 JSON



ඉගෙනීම සඳහා සැකසූ browser preview එකකි — live bank/store system එකක් නොවේ.

★ THIS MODULE'S MISSION

XAMPP මත PHP endpoint එකක් විවෘත කරන්න.

06.01 PHP කියන්නේ මොකක්ද?

PHP server එකේ ධාවනය වෙන programming language එකකි. Browser එකට PHP source file නොව output යවයි.

EXPLORER / PROJECT api.php

```php
<?php
// PHP runs on the server.
echo "Hello from PHP";
```

UTF-8 • api.php • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

index.php සහ index.html වෙනස හඳුනාගන්න.

CHECK · ප්‍රතිඵලය

PHP server-side බව තේරේ.

TIP PHP code browser JavaScript එකක් ලෙස ධාවනය නොවේ.

06.02 XAMPP ස්ථාපනය

XAMPP වල Apache, PHP සහ MySQL/MariaDB එකට භාවිතා කළ හැකියි. මේ project එක local Windows පරිගණකයක් සඳහායි.

Command Prompt _ □ ×

```
C:\Users\Student> C:\xampp\htdocs\little-star-ajax\
Start Apache + MySQL in XAMPP
```

DO IT · ඔබ කරන්න

XAMPP Control Panel විවෘත කර Apache start කරන්න.

CHECK · ප්‍රතිඵලය

localhost පිටුව පෙන්විය යුතුයි.

TIP අන්තර්ජාලයට XAMPP default settings සමඟ expose නොකරන්න.

06.03 htdocs folder

XAMPP වල සාමාන්‍යයෙන් C:\xampp\htdocs යටතේ project එක තබනවා. Folder name URL එකේ path එක වෙයි.

```
Command Prompt _ □ ×

C:\Users\Student> C:\xampp\htdocs\little-star-ajax\index.php
http://localhost/little-star-ajax/
```

DO IT · ඔබ කරන්න

Project folder එක htdocs තුළ copy කරන්න.

CHECK · ප්‍රතිඵලය

localhost/little-star-ajax/ URL එක ක්‍රියා කරයි.

TIP Project එක file:// ආකාරයට විවෘත කරන්න එපා.

06.04 PHP first response

echo සහ json_encode භාවිතයෙන් සරල JSON message එකක් යවන්න පුළුවන්. Header එක JSON ලෙස සකසන්න.

EXPLORER / PROJECT api.php

```php
<?php
header("Content-Type: application/json");
echo json_encode(["message" => "Hello!"]);
```

UTF-8 • api.php • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

mini.php file එකක් සාදා HTTP ඔස්සේ අරිනවා.

CHECK · ප්‍රතිඵලය

JSON hello message එක පෙනේ.

TIP Header යැවීමට පෙර echo/HTML නොතිබිය යුතුයි.

06.05 GET parameter

PHP වල $_GET මඟින් `?q=pencil` වැනි query value කියවයි. හැම user input එකම validate කළ යුතුයි.

EXPLORER / PROJECT api.php

```php
<?php
$query = $_GET["q"] ?? "";
echo json_encode(["query" => $query]);
```

UTF-8 • api.php • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

api.php?action=products&q=pencil URL එක පරීක්ෂා කරන්න.

CHECK · ප්‍රතිඵලය

පැන්සල් products JSON ලෙස ලැබිය යුතුයි.

TIP $_GET අගය විශ්වාස නොකරන්න.

06.06 POST JSON request

PHP `php://input` මඟින් request body එක කියවා json_decode කරනවා. Method සහ Content-Type පරීක්ෂා කිරීම වැදගත්.

EXPLORER / PROJECT api.php

```php
<?php
$data = json_decode(file_get_contents("php://input"), true);
$name = $data["product_name"] ?? "";
```

UTF-8 • api.php • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Suggestion form එකෙන් එක් item එකක් යවන්න.

CHECK · ප්‍රතිඵලය

201 response සහ success message පෙනේ.

TIP json_decode අසාර්ථක විය හැකි නිසා error handle කරන්න.

06.07 API response design

නිවැරදි responses සඳහා `ok`, `products` වගේ ස්ථාවර fields දෙන්න. Error වලදී `ok:false` හා reason message යවන්න.

EXPLORER / PROJECT api.php

```
// Success JSON:
{"ok":true,"products":[]}
// Error JSON:
{"ok":false,"error":"Invalid input"}
```

UTF-8 • api.php • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Success JSON සහ error JSON උදාහරණ දෙකක් ලියන්න.

CHECK · ප්‍රතිඵලය

Frontend එකට response පාලනය පහසු වේ.

TIP Private server error details response එකට නොදමන්න.

★ MODULE 06 CHECKPOINT

මෙම module එක අවසන්! පහත ප්‍රශ්නවලට පිළිතුරු ලියා බලා ඉදිරියට යන්න.

1. XAMPP ස්ථාපනය ගැන උදාහරණයක් දෙන්න.
2. PHP first response ඔබේම වචනවලින් පැහැදිලි කරන්න.
3. **Mini task:** XAMPP මත PHP endpoint එකක් විවෘත කරන්න.

☐ Read ☐ Run ☐ Explain ☐ Change one line

MODULE 07 / 12

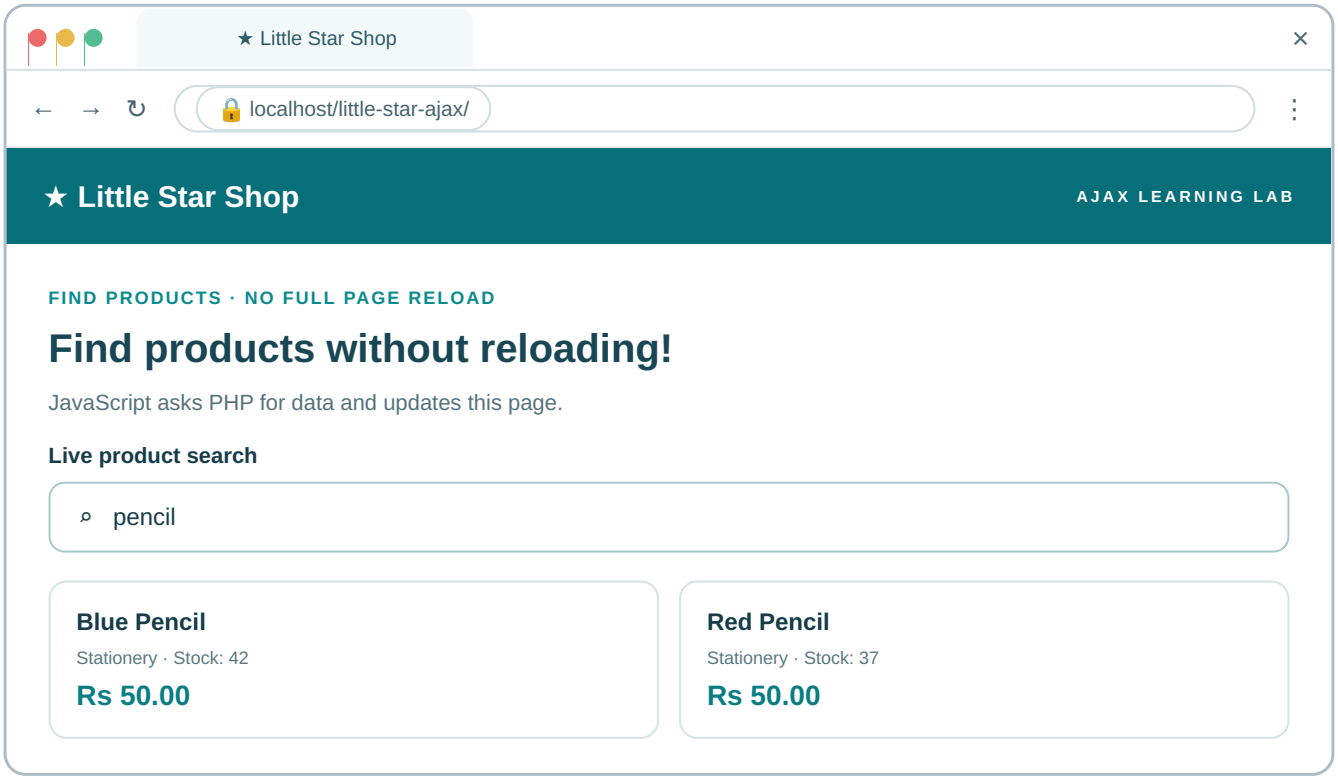
YOUR NEXT STEP

Data layer

07 MySQL සහ Database

Products සහ suggestions database එකක ආරක්ෂිතව තබමු.

01 PHP PDO → 02 SQL query → 03 MySQL



ඉගෙනීම සඳහා සැකසූ browser preview එකකි — live bank/store system එකක් නොවේ.

★ THIS MODULE'S MISSION

MySQL products table එක query කරන්න.

07.01 Database එකක් ඇයි?

Database එක project එකේ products සහ suggestions තබා ගනී. Refresh කළත් database records අතුරුදහන් නොවේ.

EXPLORER / PROJECT database.sql

```
Products table -> saved rows -> fetch later
```

UTF-8 • database.sql • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Products list එක කඩදාසියේ වගුවක් ලෙස ලියන්න.

CHECK · ප්‍රතිඵලය

Persistent data අදහස තේරේ.

TIP Browser localStorage database server එකට සමාන නොවේ.

07.02 Tables, rows, columns

Products table එකේ id, name, category, price_cents, stock_qty යන columns ඇත. එක් product එකක් row එකකි.

EXPLORER / PROJECT database.sql

```
id | name       | price_cents | stock_qty
1  | Blue Pencil| 5000        | 42
2  | A5 Notebook| 18000       | 25
```

UTF-8 • database.sql • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Product rows දෙකක් වගුවක් ලෙස ලියන්න.

CHECK · ප්‍රතිඵලය

Row සහ column වෙනස පැහැදිලි වේ.

TIP Table ID එකේ uniqueness ආරක්ෂා කරන්න.

07.03 CREATE DATABASE

phpMyAdmin > Import මඟින් sql/database.sql ධාවනය කළ විට sample database/tables සෑදේ.

EXPLORER / PROJECT database.sql

```sql
CREATE DATABASE little_star_ajax;
CREATE TABLE products (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100) NOT NULL
);
```

UTF-8 • database.sql • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Import file එකේ CREATE TABLE statements බලන්න.

CHECK · ප්‍රතිඵලය

products සහ suggestions tables සෑදේ.

TIP දැනටමත් database එකක් තිබේ නම් import කරන්න පෙර backup ගන්න.

07.04 SELECT query

SELECT query මඟින් database එකෙන් rows කියවයි. Search API එක GET route මඟින් data ලබාදෙයි.

EXPLORER / PROJECT database.sql

```sql
SELECT id, name, price_cents
FROM products ORDER BY name;
```

UTF-8 • database.sql • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

phpMyAdmin SQL tab එකේ SELECT * FROM products; ධාවනය කරන්න.

CHECK · ප්‍රතිඵලය

Sample products පෙනේ.

TIP Production API එකක SELECT * අනවශ්‍ය columns leak කළ හැක.

07.05 WHERE සහ LIKE

WHERE condition එකක්; LIKE `%pencil%` යනු text matching ය. Query එක server එකේ limit එකක් සමඟ යවයි.

EXPLORER / PROJECT database.sql

```sql
SELECT name FROM products
WHERE name LIKE "%pencil%";
```

UTF-8 • database.sql • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Pencil නම් තිබෙන rows පමණක් ගන්න.

CHECK · ප්‍රතිඵලය

Blue/Red Pencil results පෙනේ.

TIP Input එක query string එකට කෙළින් concatenate කරන්න එපා.

07.06 PDO prepared statements

PDO::prepare සහ execute placeholders භාවිතයෙන් user input SQL syntax එකෙන් වෙන් කරයි; SQL injection අවදානම අඩු කරයි.

EXPLORER / PROJECT api.php

```php
$stmt = $pdo->prepare(
  "SELECT * FROM products WHERE name LIKE :term"
);
$stmt->execute(["term" => "%pencil%"]);
```

UTF-8 • api.php • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

api.php එකේ :term placeholder සොයාගන්න.

CHECK · ප්‍රතිඵලය

Safe query pattern එක පැහැදිලි වේ.

TIP Column/table names user input එකෙන් build නොකරන්න.

07.07 INSERT suggestion

POST form එකෙන් යැවූ name/note validation පසු INSERT statement එකකට bind කරනවා. Browser එකම database credentials නොදනී.

EXPLORER / PROJECT api.php

```php
$stmt = $pdo->prepare(
  "INSERT INTO suggestions(product_name,note)"
  . " VALUES(:name,:note)"
);
$stmt->execute(["name"=>$name,"note"=>$note]);
```

UTF-8 • api.php • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Product suggestion යවා phpMyAdmin හි row බලන්න.

CHECK · ප්‍රතිඵලය

suggestions table එකට අලුත් row එකක් ලැබේ.

TIP POST request එකේ server-side validation අත්‍යවශ්‍යයි.

★ MODULE 07 CHECKPOINT

මෙම module එක අවසන්! පහත ප්‍රශ්නවලට පිළිතුරු ලියා බලා ඉදිරියට යන්න.

1. Tables, rows, columns ගැන උදාහරණයක් දෙන්න.
2. SELECT query ඔබේම වචනවලින් පැහැදිලි කරන්න.
3. **Mini task:** MySQL products table එක query කරන්න.

☐ Read ☐ Run ☐ Explain ☐ Change one line

MODULE 08 / 12

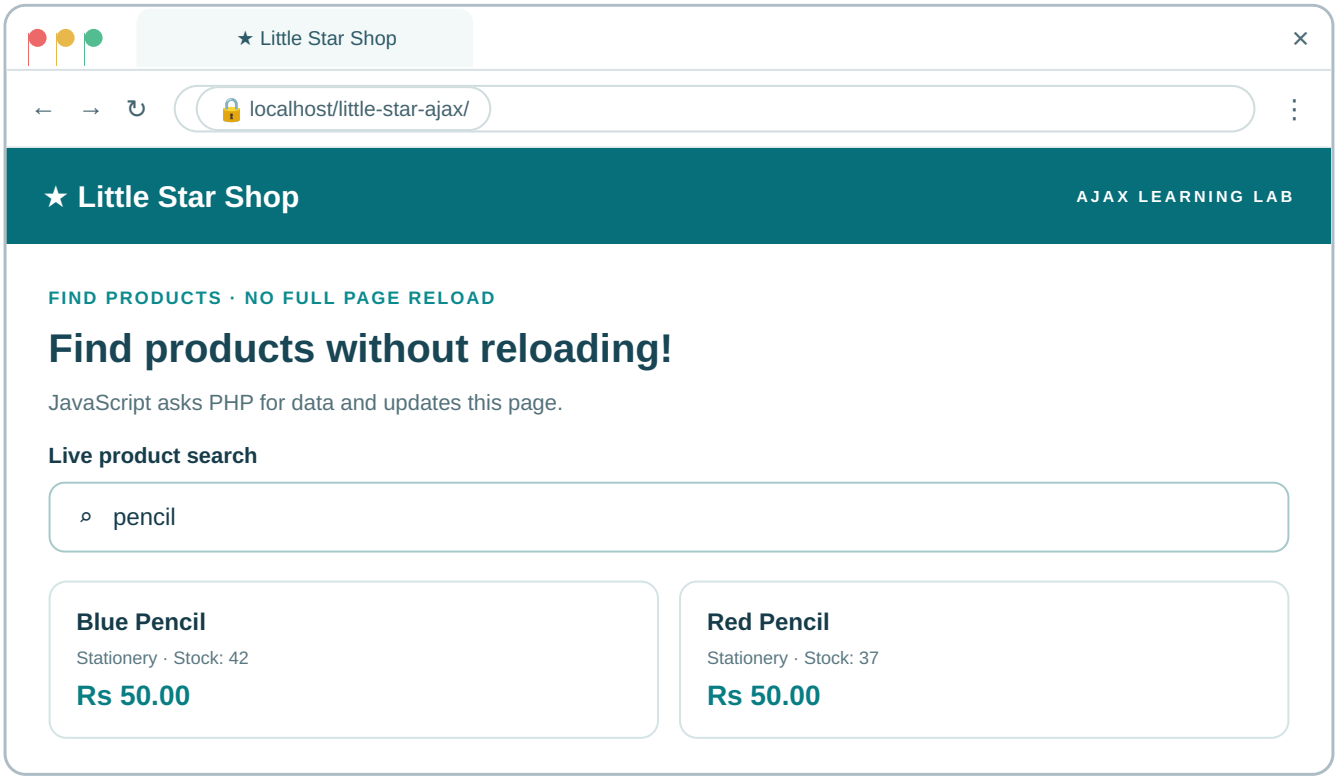
YOUR NEXT STEP

Project files

08 Little Star Shop ගොඩනගමු

දැන් HTML + CSS + JS + PHP + MySQL එකට සම්බන්ධ කරමු.

01 Project folder → 02 localhost → 03 Chrome



ඉගෙනීම සඳහා සැකසූ browser preview එකකි — live bank/store system එකක් නොවේ.

★ THIS MODULE'S MISSION

Source ZIP එකෙන් සම්පූර්ණ project එක සකස් කරන්න.

08.01 Project structure

Project root එකේ index.php, api.php, db.php, helpers.php, assets සහ sql folders ඇත. එක එක්කෙනාට වෙනම කාර්යයක්.

EXPLORER / PROJECT README.txt

```
little-star-ajax/
├ index.php
├ api.php
├ db.php
├ helpers.php
├ assets/
└ sql/database.sql
```

UTF-8 • README.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

ZIP extract කර VS Code හි Open Folder කරන්න.

CHECK · ප්‍රතිඵලය

Explorer panel එකේ files පෙනේ.

TIP Code samples වැරදි folder වලට copy නොකරන්න.

08.02 Configuration copy

config.example.php copy කර config.php කරන්න. Local database username/password මෙහි තබන්න. .gitignore එකෙන් secret settings Git වලට නොයවයි.

EXPLORER / PROJECT README.txt

```
copy config.example.php config.php
// Adjust local MySQL credentials.
```

UTF-8 • README.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

config.php file එක copy කරගන්න.

CHECK · ප්‍රතිඵලය

DB settings config.php හි ඇත.

TIP Production secrets public GitHub එකකට upload කරන්න එපා.

08.03 Import sample records

sql/database.sql හි නාම, category සහ price_cents values තබා තිබේ. මේවා අධ්‍යාපනික sample values පමණයි.

```
Command Prompt _ □ ×

C:\Users\Student> http://localhost/phpmyadmin
Import -> sql/database.sql
```

DO IT · ඔබ කරන්න

phpMyAdmin import කර products rows ගණන බලන්න.

CHECK · ප්‍රතිඵලය

Products 8ක් ලැබිය යුතුයි.

TIP Import නැවත කරන විට sample rows duplicate විය හැක.

08.04 First browser view

index.php session token එක සාදා page එක render කරයි. CSS UI එක සකසයි. JavaScript DOM එකට event listeners සම්බන්ධ කරයි.

```
Command Prompt _ □ ×

C:\Users\Student> http://localhost/little-star-ajax/
// Search + form should appear.
```

DO IT · ඔබ කරන්න

http://localhost/little-star-ajax/ විවෘත කරන්න.

CHECK · ප්‍රතිඵලය

Search box සහ suggestion form පෙනේ.

TIP Apache හෝ MySQL start කර නොමැති නම් දත්ත නොපෙන්විය හැක.

08.05 PHP API actions

api.php? action=products, stats, suggest යන routes වෙන් කරයි. හැම action එකම JSON response ලබාදෙයි.

EXPLORER / PROJECT README.txt

```
GET /api.php?action=products&q=pen
GET /api.php?action=stats
POST /api.php?action=suggest
```

UTF-8 • README.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Network tab එකේ action=stats request එක සොයන්න.

CHECK · ප්‍රතිඵලය

Product count response එක බලන්න.

TIP API method එක නොගැළපේ නම් 405 ආපසු යැවිය යුතුයි.

08.06 CSS responsive design

Grid layout එක screen width අනුව product cards පේළිවලට බෙදයි. Form labels කියවීමට පහසුව තබාගන්න.

EXPLORER / PROJECT README.txt

```
.product-grid {
  display: grid;
  grid-template-columns:
    repeat(auto-fill, minmax(190px, 1fr));
}
```

UTF-8 • README.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Chrome DevTools device toolbar on කරන්න.

CHECK · ප්‍රතිඵලය

කුඩා screen එකක cards stack වෙයි.

TIP Text දීර්ඝ කළ විට horizontal scroll ඇතිද බලන්න.

08.07 Reading source code

මූලිකව index.php බලන්න; පසුව assets/app.js; ඊළඟට api.php සහ db.php කියවන්න. Browser/server සීමාව හඳුනාගන්න.

EXPLORER / PROJECT README.txt

```
index.php  -> UI and CSRF token
assets/app.js -> AJAX and DOM
api.php    -> JSON endpoints
db.php     -> PDO connection
```

UTF-8 • README.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Files හතරෙන් input→JSON→DB ගමන ලියන්න.

CHECK · ප්‍රතිඵලය

Code responsibilities වෙන් කරන්න පුළුවන්.

TIP File එකක සියලු logic එකතු කළොත් නඩත්තු කිරීමට අසීරුයි.

★ MODULE 08 CHECKPOINT

මෙම module එක අවසන්! පහත ප්‍රශ්නවලට පිළිතුරු ලියා බලා ඉදිරියට යන්න.

1. Configuration copy ගැන උදාහරණයක් දෙන්න.
2. First browser view ඔබේම වචනවලින් පැහැදිලි කරන්න.
3. **Mini task:** Source ZIP එකෙන් සම්පූර්ණ project එක සකස් කරන්න.

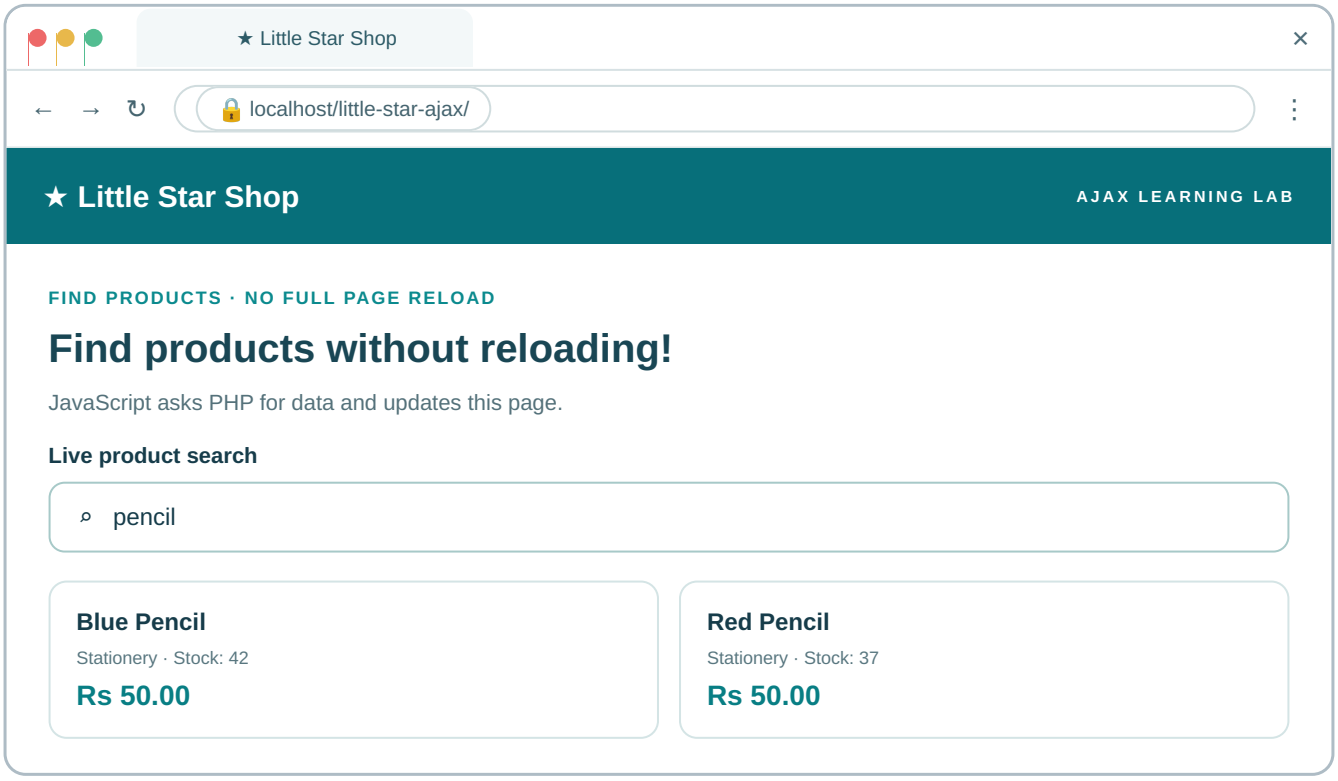
☐ Read ☐ Run ☐ Explain ☐ Change one line

Search experience

09 Live Search හදමු

Typing කරන විට data ලබාගෙන ප්‍රතිඵල අලුත් කරමු.

01 Input event → 02 fetch GET → 03 Product cards



ඉගෙනීම සඳහා සැකසූ browser preview එකකි — live bank/store system එකක් නොවේ.

★ THIS MODULE'S MISSION

pencil / notebook / zzzz කියා live search කරන්න.

09.01 Input event listener

Search field එකේ එක අකුරක් වෙනස් වූ විට input event එක සිදුවේ. ඒ සමඟ query එකේ අලුත් value ලබාගනිමු.

EXPLORER / PROJECT assets/app.js

```
1  searchInput.addEventListener("input", () => {
2    // Read input as the user types.
3  });
```

UTF-8 • assets/app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

searchInput.addEventListener("input",...) code සොයන්න.

CHECK · ප්‍රතිඵලය

Typing කළ විට listener ක්‍රියා කරයි.

TIP හැම keypress එකකටම අධික requests නොයවන්න.

09.02 encodeURIComponent

User search text URL එකේ query parameter එකක් ලෙස යවන විට encodeURIComponent භාවිතා කරනවා.

EXPLORER / PROJECT assets/app.js

```
1  const url = "api.php?action=products&q="
2    + encodeURIComponent(searchInput.value);
```

UTF-8 • assets/app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Space සහ & character තිබෙන text එකක් search කරන්න.

CHECK · ප්‍රතිඵලය

URL එක නිසි පරිදි encode වේ.

TIP Raw input එක string concatenation කර URL බිඳ දමන්න එපා.

09.03 Debounce

User typing නතර කළ පසු මිලි තත්පර 300කට පසු request යැවීමෙන් server requests අඩු කරගන්නවා.

EXPLORER / PROJECT assets/app.js

```
clearTimeout(searchTimer);
searchTimer = setTimeout(() => {
  loadProducts(searchInput.value);
}, 300);
```

UTF-8 • assets/app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

app.js එකේ setTimeout 300 හඳුනාගන්න.

CHECK · ප්‍රතිඵලය

දිගට type කරන විට requests අඩුවේ.

TIP Debounce මිලි තත්පර ගණන වැරදි ලෙස ඉතා විශාල නොකරන්න.

09.04 AbortController

පැරණි search request එක ආපසු එන්නට පෙර අලුත් request යැවූ විට පෙර request එක abort කළ හැකියි.

EXPLORER / PROJECT assets/app.js

```
if (activeRequest) activeRequest.abort();
const controller = new AbortController();
fetch(url, { signal: controller.signal });
```

UTF-8 • assets/app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

activeRequest.abort() code සොයන්න.

CHECK · ප්‍රතිඵලය

පැරණි response අලුත් results වෙනුවට පෙනීම අඩුවේ.

TIP AbortError සාමාන්‍ය search error එකක් ලෙස user ට පෙන්වන්න එපා.

09.05 DOM product cards

Products JSON එක map/forEach කර document.createElement() වලින් safe cards හදනවා.

EXPLORER / PROJECT assets/app.js

```
1  const title = document.createElement("h3");
2  title.textContent = product.name;
3  card.appendChild(title);
```

UTF-8 • assets/app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

makeProductCard function එකේ card elements ගණන් කරන්න.

CHECK · ප්‍රතිඵලය

Product name, category, price, stock පෙන්වයි.

TIP Server text සඳහා innerHTML භාවිතා නොකරන්න.

09.06 Loading feedback

fetch ආරම්භ කරන විට Searching... පෙන්වයි; success පසු results count පෙන්වයි. Error වලට හේතුවක් පෙන්වයි.

EXPLORER / PROJECT assets/app.js

```
1  searchStatus.textContent = "Searching…";
2  // On success: "2 product(s) found."
3  // On error:   "Please retry."
```

UTF-8 • assets/app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Search දෙවරක් කර status message වෙනස් වෙයිද බලන්න.

CHECK · ප්‍රතිඵලය

User ට app එකේ තත්ත්වය තේරේ.

TIP Loading message සදාකාලිකවම තබන්න එපා.

09.07 Empty results

Search එකට ගැළපෙන records 0 නම් හිස් grid එකක් හා 0 products found යන message එකක් පෙන්විය හැකියි.

EXPLORER / PROJECT assets/app.js

```
1  productsBox.replaceChildren();
2  searchStatus.textContent = "0 product(s) found.";
```

UTF-8 • assets/app.js • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

zzzz-not-found ලෙස search කරන්න.

CHECK · ප්‍රතිඵලය

0 product(s) found පෙනේ.

TIP Empty result එක server error එකක් නොවේ.

★ MODULE 09 CHECKPOINT

මෙම module එක අවසන්! පහත ප්‍රශ්නවලට පිළිතුරු ලියා බලා ඉදිරියට යන්න.

1. encodeURIComponent ගැන උදාහරණයක් දෙන්න.
2. AbortController ඔබේම වචනවලින් පැහැදිලි කරන්න.
3. **Mini task:** pencil / notebook / zzzz කියා live search කරන්න.

☐ Read ☐ Run ☐ Explain ☐ Change one line

MODULE 10 / 12

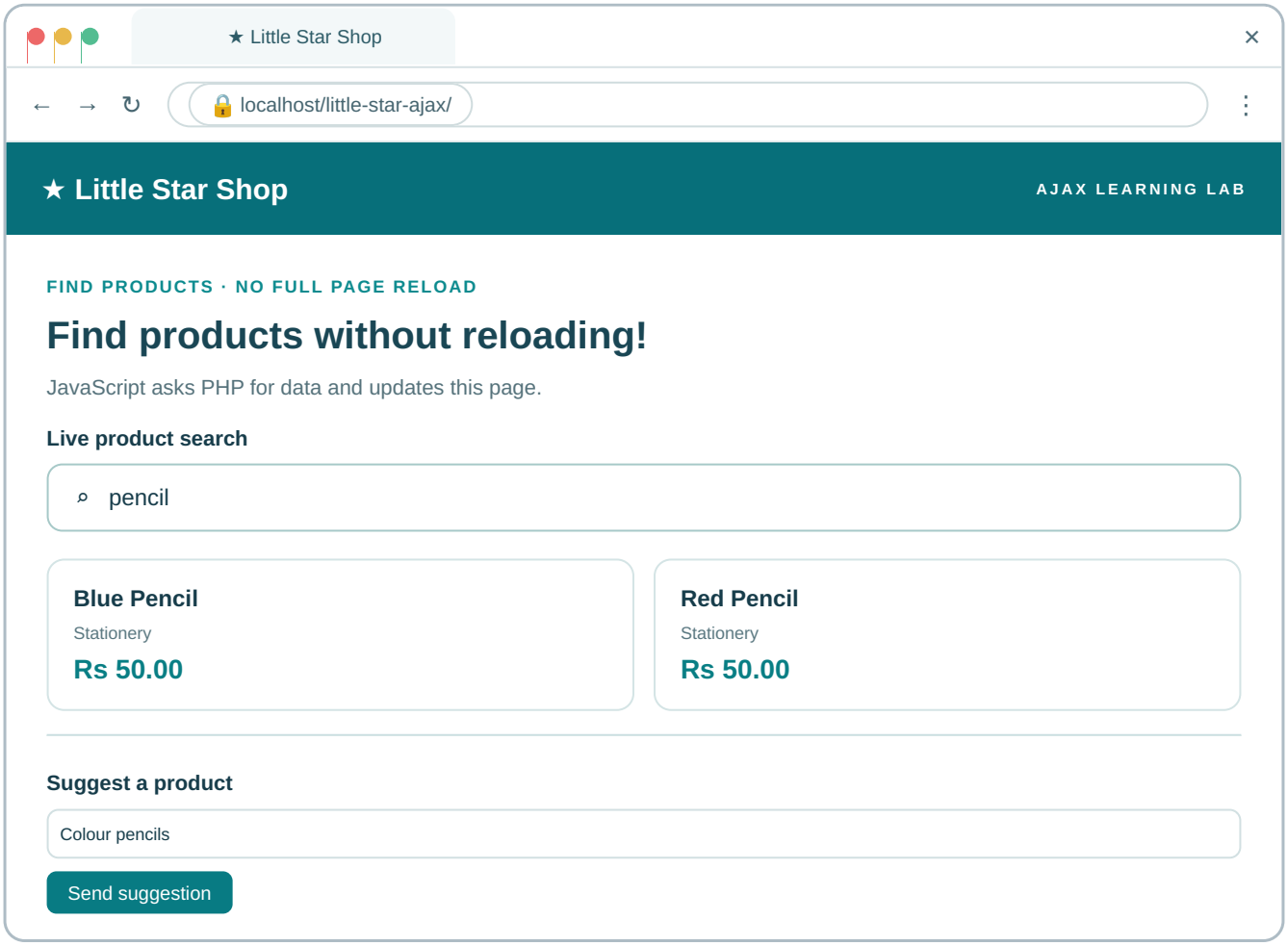
YOUR NEXT STEP

Send data

10 POST form සහ JSON API

Browser එකෙන් PHP වෙත data යවා database එකේ suggestion එකක් save කරමු.

01 Form values → 02 fetch POST → 03 SQL INSERT



ඉගෙනීම සඳහා සැකසූ browser preview එකකි — live bank/store system එකක් නොවේ.

★ THIS MODULE'S MISSION

Product suggestion form එක JSON POST මඟින් submit කරන්න.

10.01 Form submit event

Form submit වන විට event.preventDefault() මඟින් සාමාන්‍ය page navigation එක නවත්වනවා.

EXPLORER / PROJECT api.php

```
1  form.addEventListener("submit", async (event) => {
2    event.preventDefault();
3    // Send the AJAX request here.
4  });
```

UTF-8 • api.php • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Suggest new product form එකේ submit listener හඳුනාගන්න.

CHECK · ප්‍රතිඵලය

Page reload නොවී status message පෙන්වයි.

TIP මෙය server-side validation එකට විකල්පයක් නොවේ.

10.02 Read input values

JavaScript querySelector මඟින් user name/note values කියවා trim() යොදා මුල/අවසාන spaces ඉවත් කරනවා.

EXPLORER / PROJECT api.php

```
1  const payload = {
2    product_name: nameInput.value.trim(),
3    note: noteInput.value.trim()
4  };
```

UTF-8 • api.php • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Product name ලෙස Colour Pens දමන්න.

CHECK · ප්‍රතිඵලය

Payload object එකේ ඔබේ අගය යයි.

TIP User personal info එකක් demo app එකට නොදමන්න.

10.03 JSON POST headers

Content-Type: application/json සහ POST method දමා JSON.stringify(body) යවනවා.

EXPLORER / PROJECT api.php

```
fetch("api.php?action=suggest", {
  method: "POST",
  headers: { "Content-Type": "application/json" },
  body: JSON.stringify(payload)
});
```

UTF-8 • api.php • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Chrome Network tab තුළ Request Payload පරීක්ෂා කරන්න.

CHECK · ප්‍රතිඵලය

JSON fields දෙක දැකගන්න.

TIP JSON header නොදීමෙන් 415 error ලැබිය හැක.

10.04 Validate in PHP

PHP තවත් වරක් product name දිග, note දිග, Content-Type සහ JSON syntax පරීක්ෂා කරයි.

EXPLORER / PROJECT api.php

```
$name = trim((string)($payload["product_name"] ?? ""));
$errors = validateSuggestion($name, $note);
```

UTF-8 • api.php • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

එක අකුරක් පමණක් name ලෙස යැවීම උත්සාහ කරන්න.

CHECK · ප්‍රතිඵලය

Invalid value server එකෙන් ප්‍රතික්ෂේප වේ.

TIP HTML required/minlength පමණක් විශ්වාස නොකරන්න.

10.05 CSRF protection

POST request එකට session එකෙන් ලැබුණු secret CSRF token එක header එකක යවනවා. PHP token එක සසඳයි.

EXPLORER / PROJECT api.php

```
1  headers: {
2    "Content-Type": "application/json",
3    "X-CSRF-Token": csrfToken
4  }
```

UTF-8 • api.php • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

api.php එකේ hash_equals සහ X-CSRF-Token සොයන්න.

CHECK · ප්‍රතිඵලය

Token නොමැති POST එකකට 403 ලැබේ.

TIP CSRF token එක URL තුළ හෝ public log තුළ දාන්න එපා.

10.06 Save and respond

Prepared INSERT query එක සාර්ථක වූ විට PHP 201 status සහ {ok:true} යවයි. ඒකට පසුව form reset වෙනවා.

EXPLORER / PROJECT api.php

```
1  http_response_code(201);
2  echo json_encode(["ok" => true]);
```

UTF-8 • api.php • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Suggestion එකක් යවා phpMyAdmin suggestions table බලන්න.

CHECK · ප්‍රතිඵලය

New row එක දැකිය හැක.

TIP Server 201 නොලැබූ විට success ලෙස පෙන්වන්න එපා.

10.07 Submit button state

Request යවන අතර submit button එක disabled කරන්න. finally block එකේ නැවත enable කරන්න.

EXPLORER / PROJECT api.php

```
1  submitButton.disabled = true;
2  try { /* await fetch(...) */ }
3  finally { submitButton.disabled = false; }
```

UTF-8 • api.php • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Form submit කරදී button state බලන්න.

CHECK · ප්‍රතිඵලය

Duplicate clicks අඩුවේ.

TIP Button disabled කිරීම පමණක් server-side duplicates වලට ප්‍රමාණවත් නොවේ.

★ MODULE 10 CHECKPOINT

මෙම module එක අවසන්! පහත ප්‍රශ්නවලට පිළිතුරු ලියා බලා ඉදිරියට යන්න.

1. Read input values ගැන උදාහරණයක් දෙන්න.
2. Validate in PHP ඔබේම වචනවලින් පැහැදිලි කරන්න.
3. **Mini task:** Product suggestion form එක JSON POST මඟින් submit කරන්න.

☐ Read ☐ Run ☐ Explain ☐ Change one line

MODULE 11 / 12

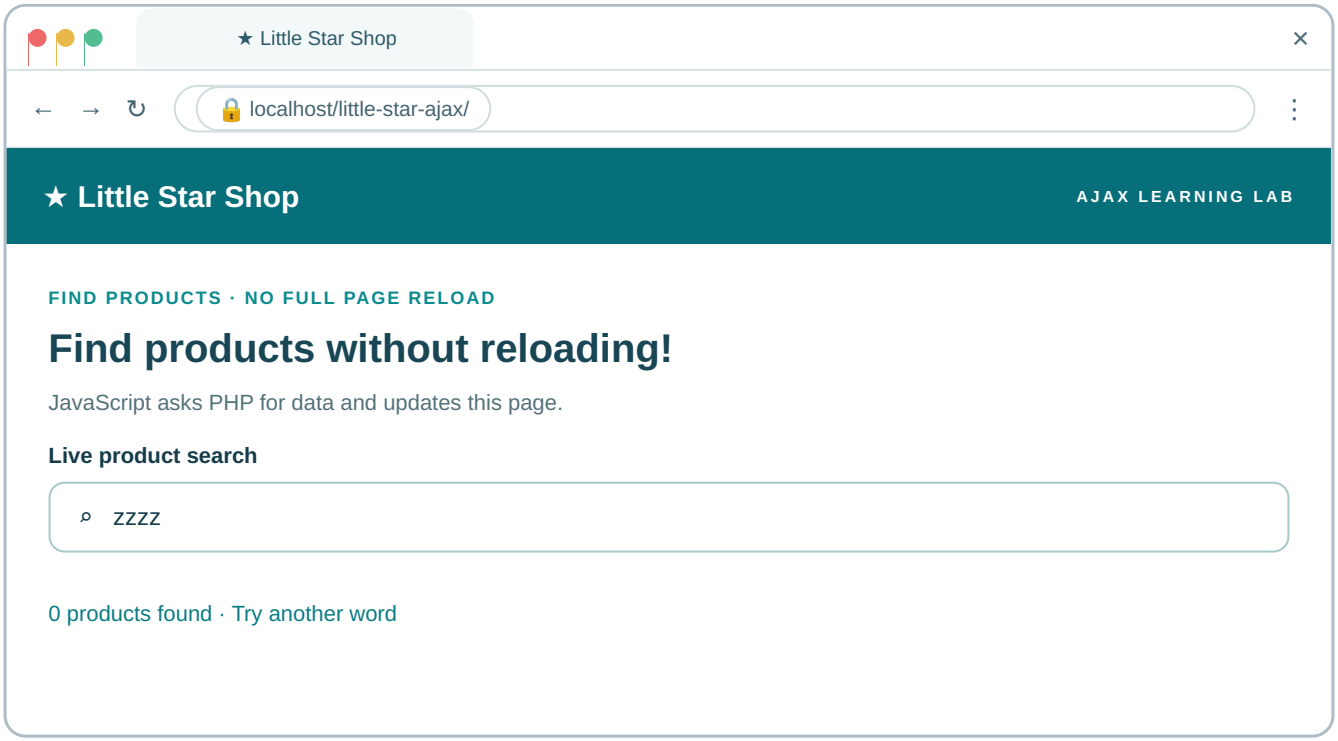
YOUR NEXT STEP

Safe beginners

11 Security සහ Privacy

AJAX එක සරල වුණත් browser input සෑම විටම විශ්වාස නොකරන්න.

01 Validate → 02 Authorize → 03 Log safely



ඉගෙනීම සඳහා සැකසූ browser preview එකකි — live bank/store system එකක් නොවේ.

★ THIS MODULE'S MISSION

Untrusted inputs හරහා security test examples කරන්න.

11.01 Never trust user input

User browser එකේ validation rules වෙනස් කළ හැකියි. PHP backend එක සෑම request එකක්ම validate කළ යුතුයි.

EXPLORER / PROJECT security.txt

```
1  // Both browser and PHP validate user input.
2  if (!isValidProductName($name)) { /* reject */ }
```

UTF-8 • security.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

helpers.php validation function ලැයිස්තුව බලන්න.

CHECK · ප්‍රතිඵලය

Browser + Server validation දෙකම ඇති බව තේරේ.

TIP Client-only check එකෙන් database ආරක්ෂා නොවේ.

11.02 SQL Injection අවදානම

SQL එකට raw user text එක concatenate කිරීම අවදානම්. PDO placeholders සමඟ prepared statements භාවිතා කරන්න.

EXPLORER / PROJECT security.txt

```
1  // BAD: "SELECT..." . $_GET["q"]
2  // GOOD: bind placeholders using PDO.
```

UTF-8 • security.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

SELECT SQL එකේ :term සහ execute array සොයාගන්න.

CHECK · ප්‍රතිඵලය

Safe SQL data binding දැකගන්න.

TIP Table/column identifiers placeholders මඟින් bind කළ නොහැක.

11.03 XSS සහ textContent

HTML special characters ඇති text DOM වෙත ඇතුළත් කරන විට textContent භාවිතා කළොත් script ලෙස ධාවනය නොවේ.

EXPLORER / PROJECT security.txt

```
// SAFE for untrusted text:
element.textContent = userInput;
```

UTF-8 • security.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Product title සඳහා textContent තෝරාගෙන තිබෙන්නේ ඇයි කියන්න.

CHECK · ප්‍රතිඵලය

Untrusted text plain text ලෙස පෙන්වයි.

TIP innerHTML තුළ unknown text දමන්න එපා.

11.04 HTTPS සහ secrets

Public website එකක HTTPS අනිවාර්ය වේ. Database password config file එකක තබා GitHub වෙත upload නොකරන්න.

EXPLORER / PROJECT security.txt

```
/config.php
// Listed in .gitignore, not GitHub.
```

UTF-8 • security.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

.gitignore file එකේ config.php entry බලන්න.

CHECK · ප්‍රතිඵලය

Local-only config secret commit නොවිය යුතුයි.

TIP Demo root/blank DB password live server සඳහා නොවේ.

11.05 Authentication & roles

Demo එකේ login නැහැ. Real app එකක් online යැවීමට login, authorization සහ staff roles අවශ්‍යයි.

EXPLORER / PROJECT security.txt

```
1  // Production requirement:
2  // check session + permissions on each endpoint.
```

UTF-8 • security.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Product editing API එක public විය යුතුද කියා සාකච්ඡා කරන්න.

CHECK · ප්‍රතිඵලය

Protected operations user/role මත රඳා පවතී.

TIP CSRF token එක authentication එකට විකල්පයක් නොවේ.

11.06 Rate limits & spam

Public suggestion endpoint එක spam ඉල්ලීම් වලින් ආරක්ෂා කිරීමට rate limiting, logging සහ moderation අවශ්‍යයි.

EXPLORER / PROJECT security.txt

```
1  // Production requirement:
2  // rate limits + moderation + audit logs
```

UTF-8 • security.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Minute එකකට requests ගණනක් සීමා කළ හැකිද සිතන්න.

CHECK · ප්‍රතිඵලය

Production hardening අවශ්‍ය බව තේරේ.

TIP Demo app එක public deploy නොකරන්න.

MODULE 11 · Safe beginners LESSONS 07 - 07 OF 07

11.07 Privacy for kids

පාඩම්වලදී සැබෑ customer names, phone numbers, passwords හෝ bank details එකතු නොකරන්න. Sample data පමණක් භාවිතා කරන්න.

EXPLORER / PROJECT security.txt

```
1  // No card data, passwords, phone numbers
2  // in this educational demo.
```

UTF-8 • security.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Form එකේ personal-info field එකක් තිබේද බලන්න.

CHECK · ප්‍රතිඵලය

Product suggestion පමණක් ගබඩා කරයි.

TIP Data minimization වඩා හොඳ ආරක්ෂාවකි.

★ MODULE 11 CHECKPOINT

මෙම module එක අවසන්! පහත ප්‍රශ්නවලට පිළිතුරු ලියා බලා ඉදිරියට යන්න.

1. SQL Injection අවදානම ගැන උදාහරණයක් දෙන්න.
2. HTTPS සහ secrets ඔබේම වචනවලින් පැහැදිලි කරන්න.
3. **Mini task:** Untrusted inputs හරහා security test examples කරන්න.

☐ Read ☐ Run ☐ Explain ☐ Change one line

MODULE 12 / 12

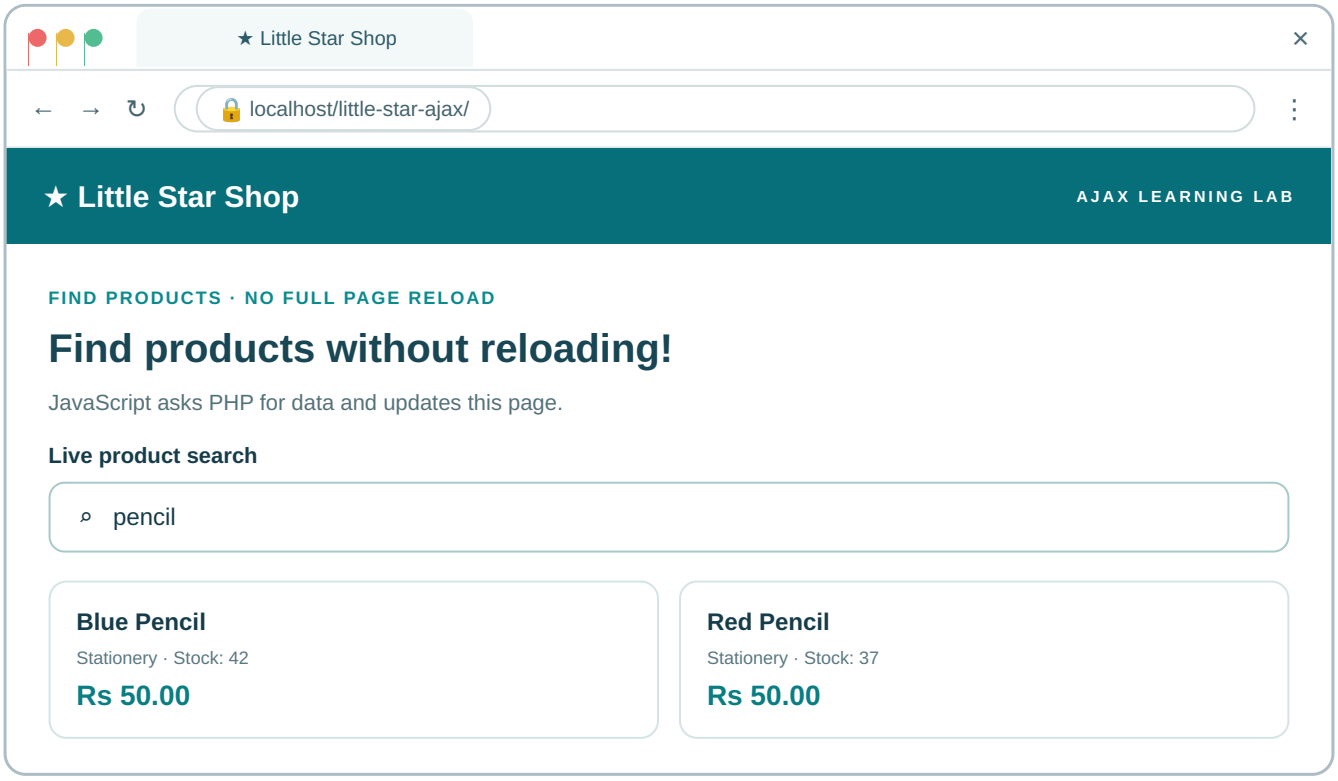
YOUR NEXT STEP

Finish well

12 Testing, Git සහ Next Steps

QA, debugging, Git හා publishing සම්බන්ධ මූලික මාර්ගෝපදේශය.

01 Test → 02 Fix → 03 Document



ඉගෙනීම සඳහා සැකසූ browser preview එකකි — live bank/store system එකක් නොවේ.

★ THIS MODULE'S MISSION

Test cases සටහන් කර ඔබේ පළමු Git commit කරන්න.

12.01 Chrome DevTools Network

F12 > Network > Fetch/XHR tab එකෙන් request URL, status, response සහ timing දැකගන්න පුළුවන්.

EXPLORER / PROJECT tests.txt

```
F12 -> Network -> Fetch/XHR
Inspect status and Response tabs.
```

UTF-8 • tests.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Pencil search කර products request එක click කරන්න.

CHECK · ප්‍රතිඵලය

Request/Response JSON හඳුනාගනී.

TIP Console error හා Network error එකම නොවේ.

12.02 Manual QA test cases

Search, empty results, invalid form, server stop, mobile layout වැනි scenarios සඳහා Expected vs Actual සටහන් කරන්න.

EXPLORER / PROJECT tests.txt

```
T01: Search pencil -> two pencils
T02: Search zzzz -> zero items
T03: Stop MySQL -> clear error
```

UTF-8 • tests.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

docs/QA_TEST_CASES.md file එක විවෘත කරන්න.

CHECK · ප්‍රතිඵලය

TC01 වැනි test case ලියන්න පුළුවන්.

TIP Happy path පමණක් නොව error path ද පරීක්ෂා කරන්න.

12.03 Automated PHP unit tests

helpers.php සඳහා PHP unit-style assertions සකස් කර තිබෙනවා. මේවා database නොමැතිවත් ධාවනය කළ හැකියි.

```
Command Prompt _ □ ×

C:\Users\Student> php tests/test_helpers.php
```

DO IT · ඔබ කරන්න

php tests/test_helpers.php command ධාවනය කරන්න.

CHECK · ප්‍රතිඵලය

PASS results 12ක් පෙනේ.

TIP Unit pass වුණා කියලා full integration pass නොවෙයි.

12.04 Git init සහ commit

Version control මඟින් ඔබේ වෙනස්කම් සටහන් කරගත හැකියි. Source code commit කරන්න, credentials නොකරන්න.

```
Command Prompt _ □ ×

C:\Users\Student> git init
git add .
git commit -m "Build first AJAX app"
```

DO IT · ඔබ කරන්න

git status, git add ., git commit commands අනුගමනය කරන්න.

CHECK · ප්‍රතිඵලය

පළමු commit එක නිර්මාණය වේ.

TIP config.php වැනි secret files Git වෙත එක් නොකරන්න.

12.05 Publish කරන්න පෙර

Demo app එකේ login, rate limit, privacy review, backup, HTTPS සහ full MySQL integration tests නොමැති නිසා public release සඳහා සූදානම් නැහැ.

EXPLORER / PROJECT tests.txt

```
Production checklist:
[ ] HTTPS   [ ] Auth & roles
[ ] DB backups  [ ] Rate limits
[ ] Logs    [ ] Full QA tests
```

UTF-8 • tests.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

Release checklist එකේ incomplete දේවල් සලකුණු කරන්න.

CHECK · ප්‍රතිඵලය

Local demo සහ production app වෙනස තේරේ.

TIP Public deployment එකට පෙර security review අවශ්‍යයි.

12.06 2026 Web development

2026 දී fetch(), async/await, JSON APIs, AbortController හා browser DevTools web development වල මූලිකයි. Framework එකක් නොමැතිව fundamentals ඉගෙනීම වටී.

EXPLORER / PROJECT tests.txt

```
fetch() + async/await + JSON
AbortController + DevTools
// Concepts remain useful in 2026.
```

UTF-8 • tests.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

මෙම concepts React / Vue apps වලත් තිබේද සිතන්න.

CHECK · ප්‍රතිඵලය

Same concepts නව frameworks වලදීත් යොදාගනී.

TIP AI-generated code නිවැරදි බව විශ්වාස කිරීමට tests අවශ්‍යයි.

12.07 ඔබේ ඊළඟ project එක

පළමු app එකෙන් පසු login සහ authorization සහිත පාසල් පොත් ලැයිස්තු app එකක් හෝ ticket tracker එකක් design කළ හැකියි.

EXPLORER / PROJECT tests.txt

```
1 Next project: Helpdesk tickets
2 Write SRS -> Sketch -> Build -> QA
```

UTF-8 • tests.txt • EGOTECHWORLD LEARNING LAB

DO IT · ඔබ කරන්න

SRS එකක් සහ flowchart එකක් සමඟ next app එක සැලසුම් කරන්න.

CHECK · ප්‍රතිඵලය

මෙම AJAX පාඩම් නැවත භාවිත කළ හැක.

TIP Features වැඩි කරන විට QA සහ backups අමතක නොකරන්න.

★ MODULE 12 CHECKPOINT

මෙම module එක අවසන්! පහත ප්‍රශ්නවලට පිළිතුරු ලියා බලා ඉදිරියට යන්න.

1. Manual QA test cases ගැන උදාහරණයක් දෙන්න.
2. Git init සහ commit ඔබේම වචනවලින් පැහැදිලි කරන්න.
3. **Mini task:** Test cases සටහන් කර ඔබේ පළමු Git commit කරන්න.

☐ Read ☐ Run ☐ Explain ☐ Change one line

PRACTICAL LAB 01 OF 08

HTML button එකකින් ආරම්භය

HTML button එක click කළ විට JavaScript වෙනස්කම බලන්න. Server එකක් නැතිව මේ මුල් පියවර කළ හැක.

FILE / lessons/01_hello_ajax.html

EXPLORER / PROJECT lessons/01_hello_ajax.html

```
<!doctype html>
<html lang="en">
<meta charset="utf-8"><title>Lesson 1 - no page reload</title>
<h1>My first JavaScript update</h1>
<button id="hello">Click me</button>
<p id="message">Waiting...</p>
<script>
  // Change part of the page without reloading it.
  document.querySelector('#hello').addEventListener('click', () => {
    document.querySelector('#message').textContent = 'Hello from JavaScript!';
  });
</script>
</html>
```

UTF-8 • lessons/01_hello_ajax.html • EGOTECHWORLD LEARNING LAB

පියවරෙන් පියවර

1. Source ZIP එකෙන් `lessons/01_hello_ajax.html` file එක විවෘත කරන්න.
2. ඉහත code එක සහ actual file එකේ සමාන lines හඳුනාගන්න.
3. Open the HTML file → click the button → observe the message.
4. ලැබුණු ප්‍රතිඵලය කඩදාසියේ ලියා තබන්න.

PRACTICAL LAB 02 OF 08

JSON data file එකක් කියවමු

JSON කියන්නේ browser එකට තේරෙන text data format එකකි. මේ පාඩමට XAMPP localhost භාවිතා කරන්න.

FILE / lessons/sample.json

EXPLORER / PROJECT lessons/sample.json

```
{
  "items": ["Pencil", "Notebook", "Eraser"]
}
```

UTF-8 • lessons/sample.json • EGOTECHWORLD LEARNING LAB

පියවරෙන් පියවර

1. Source ZIP එකෙන් `lessons/sample.json` file එක විවෘත කරන්න.
2. ඉහත code එක සහ actual file එකේ සමාන lines හඳුනාගන්න.
3. Change Pencil to Pen; refresh and fetch again.
4. ලැබුණු ප්‍රතිඵලය කඩදාසියේ ලියා තබන්න.

PRACTICAL LAB 03 OF 08

fetch() මඟින් JSON ගන්න

Fetch request එකෙන් sample.json කියවා HTML ලැයිස්තුවට පෙන්වමු.

FILE / lessons/02_fetch_json.html

EXPLORER / PROJECT lessons/02_fetch_json.html

```
<!doctype html>
<html lang="en">
<meta charset="utf-8"><title>Lesson 2 - fetch JSON</title>
<h1>Read local JSON using fetch()</h1>
<button id="load">Load items</button><ul id="items"></ul>
<script>
  document.querySelector('#load').addEventListener('click', async () => {
    try {
      // Open via http://localhost, not file:// (CORS restrictions).
      const response = await fetch('sample.json');
      if (!response.ok) throw new Error('HTTP ' + response.status);
      const data = await response.json();
      const list = document.querySelector('#items');
      list.replaceChildren();
      data.items.forEach((item) => {
        const li = document.createElement('li');
        li.textContent = item;
        list.appendChild(li);
      });
    } catch (error) {
      alert('Check your localhost server. ' + error.message);
    }
  });
</script>
```

UTF-8 • lessons/02_fetch_json.html • EGOTECHWORLD LEARNING LAB

පියවරෙන් පියවර

1. Source ZIP එකෙන් `lessons/02_fetch_json.html` file එක විවෘත කරන්න.
2. ඉහත code එක සහ actual file එකේ සමාන lines හඳුනාගන්න.
3. Inspect Network → sample.json → response.
4. ලැබුණු ප්‍රතිඵලය කඩදාසියේ ලියා තබන්න.

PRACTICAL LAB 04 OF 08

MySQL tables සකස් කිරීම

phpMyAdmin Import භාවිතයෙන් products සහ suggestions tables හදන්න.

FILE / sql/database.sql

EXPLORER / PROJECT sql/database.sql

```sql
-- Little Star AJAX lab. Import with phpMyAdmin on a local XAMPP installation.
CREATE DATABASE IF NOT EXISTS little_star_ajax
  CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
USE little_star_ajax;

CREATE TABLE IF NOT EXISTS products (
  id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  category VARCHAR(60) NOT NULL,
  price_cents INT UNSIGNED NOT NULL,
  stock_qty INT UNSIGNED NOT NULL DEFAULT 0,
  INDEX idx_product_name (name)
) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS suggestions (
  id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  product_name VARCHAR(60) NOT NULL,
  note VARCHAR(180) NOT NULL DEFAULT '',
  created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB;
```

UTF-8 • sql/database.sql • EGOTECHWORLD LEARNING LAB

පියවරෙන් පියවර

1. Source ZIP එකෙන් `sql/database.sql` file එක විවෘත කරන්න.
2. ඉහත code එක සහ actual file එකේ සමාන lines හඳුනාගන්න.
3. Run SELECT id, name FROM products;
4. ලැබුණු ප්‍රතිඵලය කඩදාසියේ ලියා තබන්න.

PRACTICAL LAB 05 OF 08

PHP API එක තේරුම් ගැනීම

GET request එකක් PHP වෙත යවලා JSON ලබාගැනීම බලන්න.

FILE / api.php

EXPLORER / PROJECT api.php

```php
<?php
declare(strict_types=1);

session_start();
require __DIR__ . '/helpers.php';
require __DIR__ . '/db.php';
header('Content-Type: application/json; charset=utf-8');
header('Cache-Control: no-store');

function reply(int $status, array $body): never
{
    http_response_code($status);
    echo json_encode($body, JSON_UNESCAPED_UNICODE | JSON_UNESCAPED_SLASHES);
    exit;
}

$action = $_GET['action'] ?? '';
$method = $_SERVER['REQUEST_METHOD'] ?? 'GET';

try {
    if ($action === 'products' && $method === 'GET') {
        $query = normaliseQuery((string)($_GET['q'] ?? ''));
```

UTF-8 • api.php • EGOTECHWORLD LEARNING LAB

පියවරෙන් පියවර

1. Source ZIP එකෙන් `api.php` file එක විවෘත කරන්න.
2. ඉහත code එක සහ actual file එකේ සමාන lines හඳුනාගන්න.
3. Open /api.php?action=stats in Chrome.
4. ලැබුණු ප්‍රතිඵලය කඩදාසියේ ලියා තබන්න.

PRACTICAL LAB 06 OF 08

Live search code කියවමු

app.js searchInput, debounce, AbortController සහ DOM cards එකට සම්බන්ධ කරන ආකාරය තේරුම් ගන්න.

FILE / assets/app.js

EXPLORER / PROJECT assets/app.js

```
'use strict';

const searchInput = document.querySelector('#search');
const productsBox = document.querySelector('#products');
const searchStatus = document.querySelector('#search-status');
const form = document.querySelector('#suggest-form');
const formStatus = document.querySelector('#form-status');
const submitButton = document.querySelector('#submit-button');
const csrfToken = document.querySelector('meta[name="csrf-token"]').content;

let searchTimer;
let activeRequest;

function makeProductCard(product) {
  const card = document.createElement('article');
  card.className = 'product';
  const title = document.createElement('h3');
  title.textContent = product.name; // textContent prevents HTML injection.
  const category = document.createElement('p');
  category.textContent = product.category;
  const price = document.createElement('p');
  price.className = 'price';
```

UTF-8 • assets/app.js • EGOTECHWORLD LEARNING LAB

පියවරෙන් පියවර

1. Source ZIP එකෙන් `assets/app.js` file එක විවෘත කරන්න.
2. ඉහත code එක සහ actual file එකේ සමාන lines හඳුනාගන්න.
3. Search pencil, then notebook, then zzzz.
4. ලැබුණු ප්‍රතිඵලය කඩදාසියේ ලියා තබන්න.

PRACTICAL LAB 07 OF 08

Validation සහ SQL security

PHP input validation + PDO prepared statements අත්‍යවශ්‍යයි.

FILE / helpers.php

EXPLORER / PROJECT helpers.php

```php
<?php
declare(strict_types=1);

function isValidProductName(string $name): bool
{
    // Count Unicode characters, so Sinhala names are supported.
    return preg_match('/\A.{2,60}\z/us', trim($name)) === 1;
}

function isValidNote(string $note): bool
{
    // Notes are optional, but cannot exceed 180 Unicode characters.
    return $note === '' || preg_match('/\A.{1,180}\z/us', trim($note)) === 1;
}

function normaliseQuery(string $query): string
{
    // Trim input. Enforce the maximum length in the API too.
    return trim($query);
}

function isSafeQuery(string $query): bool
```

UTF-8 • helpers.php • EGOTECHWORLD LEARNING LAB

පියවරෙන් පියවර

1. Source ZIP එකෙන් `helpers.php` file එක විවෘත කරන්න.
2. ඉහත code එක සහ actual file එකේ සමාන lines හඳුනාගන්න.
3. Try a blank suggestion; check status.
4. ලැබුණු ප්‍රතිඵලය කඩදාසියේ ලියා තබන්න.

PRACTICAL LAB 08 OF 08

Automated QA පරීක්ෂාව

helper functions සඳහා demo unit tests ධාවනය කර PASS output බලන්න.

FILE / tests/test_helpers.php

EXPLORER / PROJECT tests/test_helpers.php

```php
<?php
declare(strict_types=1);
require dirname(__DIR__) . '/helpers.php';
function check(bool $condition, string $name): void {
    if (!$condition) {
        fwrite(STDERR, "FAIL: $name\n");
        exit(1);
    }
    echo "PASS: $name\n";
}
check(isValidProductName('Pencil'), 'Valid ASCII name');
check(isValidProductName('සිංහල පොත'), 'Valid Sinhala name');
check(!isValidProductName(''), 'Empty name rejected');
check(!isValidProductName('X'), 'Short name rejected');
check(!isValidProductName(str_repeat('A', 61)), 'Long name rejected');
check(isValidNote(''), 'Empty note allowed');
check(!isValidNote(str_repeat('X', 181)), 'Long note rejected');
check(isSafeQuery('pencil'), 'Valid search');
check(!isSafeQuery(str_repeat('A', 61)), 'Long search rejected');
check(normaliseQuery('   book  ') === 'book', 'Trim search');
check(validateSuggestion('Notebook', '') === [], 'Valid suggestion');
check(count(validateSuggestion('x', '')) === 1, 'Invalid suggestion');
```

UTF-8 • tests/test_helpers.php • EGOTECHWORLD LEARNING LAB

පියවරෙන් පියවර

1. Source ZIP එකෙන් `tests/test_helpers.php` file එක විවෘත කරන්න.
2. ඉහත code එක සහ actual file එකේ සමාන lines හඳුනාගන්න.
3. Run php tests/test_helpers.php.
4. ලැබුණු ප්‍රතිඵලය කඩදාසියේ ලියා තබන්න.

CHROME DEVELOPER TOOLS

Request එක වැරදුණා නම් මුලින් මෙතන බලන්න

F12 ඔබලා Network → Fetch/XHR වෙත යන්න. Request එකක status, URL සහ response පෙනෙන ආකාරය පහත sample එකෙන් හඳුනාගන්න.

Network

All Fetch/XHR Doc JS CSS

Name	Status	Type	Time
api.php?action=products&q=pencil	**200**	fetch	43 ms
api.php?action=stats	**200**	fetch	28 ms
missing.json	**404**	fetch	9 ms

Response

```
{"ok":true,"products":[{"name":"Blue Pencil"}]}
```

සොයන දේවල්:

200 = OK, 404 = URL නොමැත, 500 = server error. Network error එකක් නම් Apache/connection පරීක්ෂා කරන්න. Response JSON නොවේ නම් PHP warning තිබේද බලන්න.

SECURITY CHECKLIST

AJAX app එකක් ආරක්ෂිත කරන්න ඕනේ ඇයි?

සැබෑ වෙබ් අඩවියක input එකක් browser එකෙන් යැවූ බවට විශ්වාස කරන්න බැහැ. Server එකත් වෙනම validation කළ යුතුයි.

01 Validate on server

PHP මඟින් product_name length, note length සහ method පරීක්ෂා කරන්න.

02 Prepared SQL statements

Input එක SQL string එකකට එකතු නොකර PDO parameters භාවිතා කරන්න.

03 Safe DOM output

Untrusted text සඳහා `textContent` භාවිතා කරන්න. Raw innerHTML වෙනුවට DOM nodes සාදන්න.

04 CSRF + HTTPS

POST සඳහා CSRF token එක verify කරන්න. Public site එකේ HTTPS භාවිතා කරන්න.

05 Keep secrets out of Git

`config.php` තුළ database password තබා .gitignore එකට ඇතුළත් කරන්න.

06 Production is different

Demo එක public දැමීමට පෙර login, authorization, rate limits, logs, backups සහ security tests කරන්න.

QUALITY ASSURANCE

Test Cases — ඔබේ website එක නිවැරදිද?

Expected සහ Actual වෙන් කර සටහන් කරන පුරුද්ද software tester කෙනෙකුට අත්‍යවශ්‍යයි.

ID	පරීක්ෂාව	Expected	Result
T01	Open app page	Search form පෙනේ	☐
T02	Search pencil	Matching cards	☐
T03	Search zzzz	0 products found	☐
T04	Stop Apache	Friendly error	☐
T05	Invalid form	Validation error	☐
T06	Valid suggestion	Success JSON / 201	☐
T07	Mobile width	Responsive cards	☐

```
Command Prompt                                    _ ☐ ×

C:\Users\Student> php -l api.php
php -l index.php
php tests/test_helpers.php
node --check assets/app.js
```

QA note:

Unit tests සමත් වීමෙන් මුළු system එකම නිවැරදි බව ඔප්පු වෙන්නේ නැහැ. XAMPP/MySQL සමඟ පියවරෙන් පියවර integration tests ද කරන්න.

FINAL WORKSHOP

දින 7ක AJAX practice plan

HTML + Browser Lesson 01, button + DOM test	DAY 1
JavaScript basics Events, variables, textContent	DAY 2
fetch() + JSON Read sample.json and check network	DAY 3
PHP + XAMPP Localhost, JSON response	DAY 4
MySQL + CRUD ideas Import SQL and query products	DAY 5
AJAX full app Search, suggestion and error UI	DAY 6
QA + Git Test cases, notes, safe commit	DAY 7

ඔබේ අවසාන challenge එක

“Blue Pencil” වෙනුවට අලුත් product එකක් database එකට ඇතුළත් කර, search box එකෙන් එය සොයාගන්න. Request සහ response දෙක Network tab එකෙන් පරීක්ෂා කරන්න.

OFFICIAL REFERENCE

ඉදිරියට කියවන්න

MDN Fetch API

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API

MDN Using Fetch

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch

PHP PDO prepared statements

https://www.php.net/manual/en/pdo.prepared-statements.php

OWASP CSRF guidance

https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html

අවසාන සටහන

මෙම PDF එක training සඳහා නිර්මාණය කර ඇත. Diagram සහ browser previews අධ්‍යාපනික illustrations වන අතර, full source code වෙනම ZIP එකේ ඇත. ඔන්ලයින් දත්ත රැස් කිරීමකට පෙර privacy, security, backups සහ deployment QA සලකා බලන්න.