|  |  |  |
| --- | --- | --- |
| **Author** | **Reviewer** | **Approver** |
| BaoHT5 |  |  |

**Exercise: Building a Simple Expense Management App APIs Using GitHub Copilot**

## **1. Objective**

* Develop a simple, end-to-end **Expense Management App** APIs.
* Leverage **GitHub Copilot** to accelerate development, generate code, and assist with troubleshooting.
* Implement CRUD operations (view, add, edit, delete) for expense items.
* Calculate and display daily, weekly, and monthly expense totals.

## **2. Problem Statement**

Tracking personal or team expenses is a fundamental requirement in both business and personal finance.  
 Building an expense management application from scratch can be challenging due to the need for secure authentication, data management, and summary calculations.  
 This assignment requires you to build, and document the APIs for the expense manager with GitHub Copilot as your AI programming assistant.

## **3. Inputs / Shared Artifacts**

* API Details and DB structure Provided
* Use any tech stack of your choice (Node.js, Python, Java, etc.).
* **Access to GitHub Copilot** in your IDE for code suggestions and troubleshooting.

## **API Details**

|  |  |  |
| --- | --- | --- |
| **Method** | **Endpoint** | **Description** |
| POST | /api/expenses | Add a new expense |
| GET | /api/expenses | View all expenses |
| GET | /api/expenses/:id | View a single expense by ID |
| PUT | /api/expenses/:id | Edit/update an expense by ID |
| DELETE | /api/expenses/:id | Delete an expense by ID |

## **Table Structure**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| id | INTEGER (PK, auto-increment) | Unique ID for each expense |
| title | TEXT | Short title/description of the expense |
| amount | REAL | Expense amount |
| category | TEXT | Category like Food, Travel, Bills |
| date | DATE or TEXT | Date of the expense |
| notes | TEXT | (Optional) Additional notes |
| created\_at | DATETIME | Timestamp when expense was created |
| updated\_at | DATETIME | Timestamp when expense was last updated |

## **4. Expected Outcome**

* **A working expense management app** APIs with:
  + APIs to view, add, edit, delete expenses (CRUD)
* Documentation showing how GitHub Copilot supported your workflow
* Demo recording of the full flow

## **5. Concepts Covered**

* **API Development:** Authentication, backend, DB
* **CRUD Operations:** Add, edit, delete, and view items
* **Time-Based Aggregation:** Calculating daily, weekly, and monthly totals
* **Prompt-Driven Coding:** Using GitHub Copilot for code generation and debugging
* **AI-Assisted Development:** Integrating Copilot into an efficient programming workflow

## **Example: Step-by-Step Instructions**

*(Topic Example: Simple Task Tracker App Using GitHub Copilot — for practice only. For your assignment, build an expense app.)*

|  |  |
| --- | --- |
| **Step** | **Description** |
| **Step 1** | **Scaffold Your Project**  Initialize an API project (e.g., Node/Express backend, and SQLite or MongoDB). Use Copilot to help scaffold routes. |
| **Step 2** | **CRUD for Items**  Use Copilot to implement REST endpoints (add, edit, delete, view expenses). |
| **Step 3** | **Test and Polish**  Test all features; ask Copilot for improvements or bugfixes. |
| **Step 4** | **Document and Record**  Prepare a screen recording (max 5 minutes) showing:  - Viewing, adding, editing, deleting expenses  - Summarize how Copilot helped you (in a short README or PDF). |

## **Final Submission Checklist**

**Screen Recording (max 5 minutes):**

* APIs - CRUD operations, and summary features.
* Mention where GitHub Copilot was used.

**Supporting Files:**

* Source code archive or GitHub repo link
* Short documentation (README/PDF) with:
  + Copilot prompt examples or screenshots
  + Brief summary of experience, challenges, and AI assistance
  + Key takeaways

**Note for Mentee:**  
 You must invent your own sample data and users for this assignment.  
 Focus on how GitHub Copilot accelerates your development, improves code quality, and helps with troubleshooting.  
 Document your learning and improvements as you iterate on the project.