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

**Assignment: Build a Personal Blog with Login (Leverage ChatGPT, Figma, Replit)**

## 1. Objective

* Build a full stack **Personal Blog** web application with login functionality.
* Use **ChatGPT** for ideation and planning, **Figma** for UI/UX design, and **Replit** for coding and development.
* No starter code, assets, or datasets are provided; create and use your own mock data.

## 2. Problem Statement

A personal blog is a foundational web project for learning full stack development.  
However, building one from scratch can be overwhelming for beginners, especially without guidance on UI/UX, backend logic, and best coding practices.  
This project aims to teach you how to build a personal blog with secure login and CRUD operations, while using AI tools (ChatGPT, Figma) to enhance productivity and creativity.

### **Development Phase**

**Problem Statement (User Stories):**

* The project must address two core user stories, guiding the implementation and feature scope:
  + **As a user, I want to register and log in so I can access my blog dashboard.**
  + **As a user, I want to view a list of blog posts so I can browse content.**
* All implementation (including mock data) must be built from scratch.

### **Testing Phase**

**Problem Statement (Key Test Scenarios based on User Stories):**

* The project must include at least four key test scenarios to validate core functionalities:
  + **Login Test:** Verify that only valid users can log in and access the dashboard.
  + **Post Management Test:** Ensure that users can successfully view posts.
  + **Access Control Test:** Check that unauthenticated users cannot access protected routes or view posts.

## 3. Inputs / Shared Artifacts

* + You must create your own blog post data (can be mock/fake data in code).
  + All designs must be created from scratch (in Figma) or choose existing design from internet.
  + All code must be written from scratch (in Replit or your local IDE).
  + Follow **User Stories in Problem Statement**

## 4. Expected Outcome

* Working features based on user stories
* Write at least 5 test cases
* Execute tests and record results (pass/fail)
* Submit test document (Google Doc or Excel)
* Documentation and demo recording as per instructions.
* Proficient use of AI tools

## 5. Concepts Covered

* **Full stack Web Architecture:** Frontend, backend, API.
* **User Authentication:** Mock/simple login and session handling.
* **CRUD Operations:** View blog posts.
* **UI/UX Prototyping:** Designing wireframes in Figma.
* **AI Tools for Productivity:** Using ChatGPT for code scaffolding, bug fixing, and feature ideation.

## Example: Step-by-Step Instructions

|  |  |
| --- | --- |
| **Step** | **Description** |
| **Step 1** | **Plan Your Features with ChatGPT**  Use ChatGPT to brainstorm features (e.g., "What features should a personal blog have?"). Ask ChatGPT to outline basic data structure for blog posts and users. |
| **Step 2** | **Design the UI in Figma, Readdy AI**  Sketch wireframes for login, dashboard, post list, post detail, add/edit forms. Export key screens as images for documentation. (Optional) |
| **Step 3** | **Set Up Project on Replit**  Create a new project (React for frontend, Node/Express or FastAPI for backend, or use Replit’s built-in options). Scaffold the frontend and backend structure. |
| **Step 4** | **Implement Authentication (Mock/Simple)**  Create login form (email/password; hard-code a user in backend or use in-memory check). On login, show dashboard; otherwise, keep user on login screen. |
| **Step 5** | **CRUD Operations**  Backend: Use mock data as arrays (blog posts). REST API: /posts with GET Frontend: Fetch posts, show lists, buttons. |
| **Step 6** | **Use ChatGPT for Support**  Example prompts: "How do I create a protected route in React?" "Help me write Express.js CRUD endpoints for blog posts." |
| **Step 7** | **Test and Polish**  Add basic error handling (e.g., “Post not found”). Style the app to match your Figma design. Record a 3–5-minute demo video showing all core features. |
| **Step 8** | **Prepare Submission**  Include: Source code (as Replit/GitHub link), Figma screenshots or links, README or PDF with summary, key ChatGPT prompts, screenshots, challenges, and solutions. |

## Final Submission Checklist

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

* Walk through Figma design (if have)
* Feature planning, core UI, login, and CRUD flows.
* Narrate how ChatGPT and Figma supported your process.

**Supporting Files:**

* Project source code (Replit or GitHub link)
* Figma screenshots or links
* Short documentation (README/PDF):
  + Example ChatGPT prompts and how they helped
  + Main problems/challenges and how you solved them (with or without AI)
  + Key takeaways

**Note for Mentee:**  
 For your assignment, follow the above steps but apply them to a **Personal Blog with Login**.  
 You must invent your own mock blog data, design in Figma, and code from scratch with AI tool support.