**Secure To-Do List Application with User Authentication and Task Management**

**Midpoint Report**

**Author – Pouria Ghasemi**

1. Introduction
2. Original Plan and Methodology
3. Current Progress
4. Corrective Action
5. Short Summary

Word count: 2175

***1 – Introduction***

*The midpoint report is a critical part of the development cycle. It allows one an opportunity to reflect on how the project has been progressing so far and if any changes are required for the project to be delivered on schedule. In this report, I will outline the original plan of development for the application and what has been achieved so far. This will allow us to compare the two and identify potential suggestions to change or correct the plan. I will then suggest a new plan going forward in order to rectify any issues found within the first half of the project development.*

***2 - Original Plan and Methodology***

*The original objective of this project was to design and implement a secure, user-friendly web-based application that allows individuals to efficiently manage their personal tasks. This project aims to address the need for a task management system that not only helps users organise their tasks effectively and also ensures data privacy and security through reliable user authentication methods.*

*Listed below are the original functionalities that had been planned to be implemented.*

***Core Functionalities****:*

*User Authentication:*

* *A secure user authentication method must be implemented which allows user registration and login to ensure only authorised users access their data.*

*CRUD Functionality:*

* *The application must allow users to Create, Read, Update and Delete (CRUD)(ref 3) tasks easily.*

*Database Management:*

* *A secure database must be designed and implemented to store user information and tasks. The database must only be accessed by authenticated users.*

***Desirable Functionalities****:*

*Task Prioritisation.*

* *Allow users to manage tasks by priority (high, medium, low) to help users focus on main tasks.*

*Due Dates and Deadlines:*

* *Ability to add due dates and deadlines to tasks to visually highlight upcoming or overdue tasks.*

*Task Filtering and Sorting:*

* *Provide option to filter and sort tasks based on priority or due date.*

***Optional Objectives****:*

*Email Notification:*

*- Allow for set up of email notifications to remind users of upcoming or overdue tasks.*

*Task Categories:*

* *Allow for tasks to be organised into customer categories (school, work etc)*

*Statistics and insights:*

* *Provide users with statistical insights into their task management habits such as completion rate or time spent.*

*Methodology:*

*The original development methodology that was to be followed was Agile development methodology, specifically with the SCRUM framework. The essential features should be developed first, then adding the desirable and optional features later. This was to be done in Sprint format. This methodology started with a user story stated below:*

*“As a user, I want to create an account and securely log into my to-do list app so that I can access my personal list and manage my tasks privately.”*

*A product backlog was created from the functionalities specified for the app:*

*Product Backlog:*

*Essential: User registration and login, CRUD functions for tasks, data validation and security*

*Desirable: Task prioritisation, task due dates, secure session management*

*Optional: Email notifications, task categories, statistical insights*

*At the beginning of each sprint, a set of features will be selected from the product backlog which will be worked on and fully implemented, for example, setting up user registration. Each sprint will last between 2-4 weeks. This will allow for a measurable way to track progress and adjust aims and goals as needed*

*A weekly review will be implemented as a way to check-in with the project report and to monitor progress. It will be used to reflect on what was accomplished during the week, identify challenges and plan the next steps for the upcoming week*

*Sprint review:*

*At the end of each sprint, the completed features will be thoroughly tested and reviewed based on the feedback from these tests. This review process ensures that each part of the application works as intended, meets the project objectives and adheres to quality standards. During the review, any issues, bugs, or inconsistencies will be documented and addressed in the following sprint.*

*Technology stack:*

*Front End: HTML, CSS, JavaScript (React library)*

*HTML (Hypertext Markup Language) will be the backbone of the web page structure. It will define the content and the layout of the applications user interface, enabling the arrangement of elements such as forms, buttons and lists. CSS (Cascading Style Sheets) will be used to style the HTML elements. This will allow for customisation of the colours, layout and overall design of the application, which will be used to ensure a clean, user-friendly and responsive interface.*

*Back End: Node.js*

*This project will use Node.js as the language to connect the front-end system to the back end. Node.js is a JavaScript runtime environment (ref 1) which allows JavaScript code to be run on the server side of a system. By using Node.js for the backend, it will streamline the development process as the same language (JavaScript) will be used for both front end and back-end systems.*

*Database: MySql server*

*MySQL is the world most popular data management system which supports structured data storage. MySQL allows for well organised storage of the users data and tasks with clear relationships between tables. MySQL supports SQL (Standard Query Language) which provides important database query functions such as retrieve, update and delete. These functions are vital for the CRUD of tasks within the application.*

*IDE: Visual Studia Code, GitHub*

*The Integrated Development Environment that will be used for this project will be Visual Studio Code. A powerful and lightweight IDE created by Microsoft, with extensive support for JavaScript and Node.js which makes it well suited to this project. All code will be uploaded to a Github repository to aid in version control and collaboration. Github will contain a history of all changes made during development which could aid in bug fixes by reverting to older versions.*

*Original work plan with estimates:*

*Sprint 1: Project Setup*

*Task 1 – Dev environment setup*

*Start Date: Week 1 Day 1*

*Duration: 2 Days*

*Prerequisites: None*

*- Set up Development Environment and initialise project repository.*

*Install all required software (Node.js, MySQL, Git) and set up VS Code as primary IDE. Initialise code repository on Github.*

*- Create project file structure on local system*

*- Configure Github repository with README file.*

*Sprint 2: Database Design and Setup*

*Task 2 – Database schema*

*Start Date: Week 1 Day 3*

*Duration: 3 Days*

*Prerequisite: Completion of Sprint 1*

*- Create database structure using UML design with tables for users and tasks. Design fields for each table attribute (e.g., user ID, email, hashed password for users; task ID, title, description, due date, priority, and user ID for tasks). Define relationships between tables.*

*Task 3 – MySQL setup*

*Start Date: Week 1 Day 6*

*Duration: 2 Days*

*Prerequisites: Completion of task 2*

*- Install and configure MySQL. Begin to create database and set up user and task tables*

*Sprint 3: User Authentication Function*

*Task 4*

*Start Date: Week 2 Day 1*

*Duration: 7 days*

*Prerequisites: Database setup is complete*

*- Create backend function for user registration (POST /register). Set up REST API endpoint and validate user email and passwords and store data in database.*

*- Create backend function for user login (POST /login). Set up REST API endpoint for login requestions. Check email and password against database and verify login.*

*- Build Frontend form for registration and login. Connect to backend API. Implement frontend input handling.*

*Sprint 4: Task Management CRUD Functions*

*Task 5*

*Start Date: Week 3 Day 1*

*Duration: 14 days*

*Prerequisites: Database setup and User authentication function setup*

*- Create backend API for task creation function (Post /tasks). Include create and delete function. Validate that only task owner can modify tasks.*

*- Build Read and Update functions.*

*- Create frontend REACT components for CRUD functions using forms.*

*- Test CRUD functions to ensure tasks are modified correctly. Complete security testing.*

*Sprint 5: UI Design*

*Task 6*

*Start Date: Week 5 day 1*

*Duration: 8 days*

*Prerequisites: CRUD Functionality fully implemented*

*- Use CSS to create design of application. Ensure consistency in style of application across all pages.*

*Sprint 6: Security Implementation*

*Task 7*

*Start Date: Week 6 day 2*

*Duration: 8 days*

*Prerequisites: Completion of user authentication and CRUD API*

*- Add CSRF tokens for form submissions and implement server side input validation*

*Sprint 7: Extra features*

*Start Date: Week 7 day 1*

*Duration: 14 days*

*Prerequisites: Full CRUD functionality*

*- Implement task priority level and task category features. Update backend code to support new formats.*

*- Add frontend code to display new functions.*

*- Complete test for new features*

*Sprint 8: Testing and Finalisation*

*Start Date: Week 9 day 1*

*Duration: 7 Days*

*Prerequisites: completion of all features*

*- Complete Unit tests for all features.*

*- Complete security testing (e.g SQL injection, XSS)*

*- Complete application performance tests.*

*- Write up evaluation*

***3 – Current Progress***

*In this section of the report, I will go through the actual development cycle so far.*

*All Objectives and functionalities have been kept the same and development has started of the essential objectives; User authentication, CRUD functionality, Database management.*

*Methodology:*

*The original methodology of development of the application has been kept the same and followed; Agile methodology with the SCRUM framework. There was delays to start the first sprint due to a lot of pre planning and research, due to this the start of sprint 1 had to be pushed back. Howeverr, all user stories and the product backlog has been kept the same.*

*Sprint Plan:*

*The structure of the Sprints has been kept the same, with a Sprint starting with selection from the product backlog. The length and contents of the Sprints have however changed.*

*Weekly Review:*

A personal weekly review has been started digitally on the progress of the project. It has been useful to keep track of high-level objectives and quantify the time spent on each functionality.

Sprint Review:

The Sprint review strategy has merged with the weekly review to keep both high level objectives and more specific tasks in one location.

*Technology Stack:*

*Front End: JavaScript(React Library), JSX, CSS*

*The use of HTML has been replaced with JSX in the technology stack. JSX is an extension for JavaScript that allows you to write HTML-like code directly within React components. It integrates structure and logic together allowing for the layout and behaviour of the app to be on the same page. This simplifies the development of the app as it does not require a separate HTML page.*

*Back End: Node.js*

*Node.js has continued to be used effectively in the development of the app. It has help been useful in creating the JavaScript framework for the backend. Node has simplified the coding of the application as the same JavaScript framework/language has been used in the front end and backend.*

*Database: MySQL server*

*MySQL has functioned well so far to service the database needs of the application. It has been simple to host the database server on my local machine to complete quick tests of functionalities. I believe that I will continue to use this database system for the duration of development. However, during research on databases, I other alternative data management systems which are newer and more contemporary within the industry such as MongoDB and PostgreSQL. These alternatives might be useful to explore in the future*

*IDE: VS Code, GitHub*

*VS Code has continued to be the primary IDE for the project. I have found it to be extremely useful as it comes built in with extensions for React and Node.js which have streamlined the development process. GitHub has been essential for version control and tracking of changing and updates.*

***4 – Corrective Action***

*While initial delays have affected the original timeline and sprint plans of the application, development has been making progress.*

*Going forward, the main corrective actions to take are adjustments to the sprint lengths and content. I have begun to use Monday.com as a sprint tracker to simplify and streamline the sprint process.*

*The focus for the next few weeks will be:*

1. *Completing testing and debugging to ensure functionalities*
2. *Refining UI for responsiveness and user friendly design*
3. *Preparing documentation for final report*

***5 – Short Summary***

*In this document, I have evaluated the current progress of the “To-Do List Application with User Authentication” and reflected on the development process so far. Comparing the original objectives and methodologies with the actual progress allowed me to understand the short comings of the development process so far and to amend these short comings. The Agile methodology has been followed; however some adjustments have been made to the sprint lengths and contents. Moreover, the Monday.com application will be implemented into the development cycle to better track the sprints. The technology stack, including React, Node.js, and MySQL, has been effective, though alternative tools have been considered for future exploration. Corrective actions focus on ensuring testing, debugging, UI refinement, and documentation in the coming weeks to meet project objectives and deliver the application on schedule.*