VIETNAM NATIONAL UNIVERSITY – HCM

INTERNATIONAL UNIVERSITY

Final Report

Topic: Online Weekly Task/Event Management, Note-taking and Managing System

Project's Name: EasyDay

Lecturer: Assoc. Prof. Nguyễn Văn Sinh

Department: School of Computer Science and Engineering

Course: Web Application – IT093IU

No.	Name	Student ID	Contribution
1	Nguyễn Toàn Phúc	ITITIU21093	100%

Table of Contents

List of Figures

I. Chapter 1: Introduction

The “EasyDay” web application is designed to help users manage tasks/events, and notes in a simple and organized way. It provides a weekly task and event planner, a note-taking and organizing system, and a user-friendly interface. Unlike many existing calendar tools that lack effective task management and note-taking features, EasyDay combines these functionalities into one convenient platform.

This application aims to make productivity easier by keeping tasks/events, and notes in one place. It is suitable for a variety of users, including students and professionals, who need a simple solution to stay organized. Future updates will add features like reminders, customization options, and collaboration tools for shared tasks/events.

II. Chapter 2: Background

2.1 Project Background

In the development of "EasyDay," we have leveraged a range of frontend and backend technologies, a database for user and system data management, and a systematic note taking and managing approach. By the end of this report, readers will gain an understanding of the project’s objectives, features, technologies, and planned future developments.

2.1.1 Project Scopes

- **User Account Management:**
  - Secure login and registration system with user credential storage.
  - Basic profile features allowing for personalized user settings.
- **Core Features:**
  - Vertical weekly calendar interface
  - Task/Event management with the ability to add, repeat, edit, and delete tasks/events.
  - Note-taking and note management with the ability to display, add, edit, and delete notes.
- **Data Management:**
  - Secure storage of user data, tasks/events and notes in a database.
  - Backup and restore functionality to ensure data integrity.
- **Future Expansion (Planned Features):**
  - Improve note-taking UI
  - Collaboration features to allow multiple users to manage shared notes and tasks/events.

2.1.2 Project Objectives:

The primary objectives of this project include:

- **Simplify Task, Event and Note Management:** Develop an intuitive and easy-to-use platform that enables users to efficiently manage tasks/events, and notes all in one place, offering a streamlined solution for daily organization.
- **Design a User-Friendly Interface:** Create a clean, user-centric interface with a vertical weekly calendar view, making it simple for users to add, edit, and manage tasks/events, and notes in a seamless manner.
- **Ensure Secure User Account Management:** Implement a secure login and registration system with user credential storage, and provide personalized profile features to help users customize their experience.
- **Establish Reliable Data Management:** Develop a robust backend system to securely store and manage user data, including tasks/events, and notes, with backup and restore functionality to ensure data integrity.

2.2 User Roles & Permissions

At present, the roles are streamlined to include basic user roles, with potential for expanded administrative roles in future updates.

- **Users:**
    - Registered users can access all core functionalities, including calendar management, task management, and project management.
    - Users can add, edit, and delete tasks/events and notes.
    - Users have access to their profile page, where they can view and adjust personalized settings.
- **Administrator:**
    - Administrators hold the highest level of access within "EasyDay," enabling them to manage, maintain, and monitor the entire system.
    - Their responsibilities include overseeing user accounts and ensuring system optimization.
    - The admin role includes tools for:
        - Account Management: Admins can create, modify, and delete user accounts, addressing any issues users may face and maintaining community standards.
        - System Monitoring: Admins are equipped with basic tools to monitor system performance and ensure smooth operation.

- Maintenance: Admins handle the critical tasks/events of regular updates and system maintenance to ensure the platform runs efficiently.

III. Chapter 3: Project approach

3.1 Work Breakdown Structure

The "EasyDay" web application project is structured into five essential phases: **Planning & Analysis, Design** and **Development & Integration**. Each phase builds on the previous one, ensuring a smooth and efficient workflow toward project completion.

3.1.1 Planning & Analysis

The Planning & Analysis phase establishes the project's foundation by identifying objectives, scope, and user needs. This phase ensures the developer has a clear vision for "EasyDay" and defines essential features for initial development.

- **Define Project Scope & Objectives:** Clearly outline the goals, primary features, and overall vision for "EasyDay" as a comprehensive task, event, and note management platform with a user-friendly interface.
- **Define User Roles & Permissions:** Identify key user roles, such as standard users and administrators, and define permissions for accessing and managing features like task creation, note editing, and event planning.
- **Feature Prioritization:** List and prioritize features for the Minimum Viable Product (MVP) and future updates. Focus initially on secure login, a weekly calendar interface, task/event management, and basic note-taking functionality, with future enhancements like reminders and collaboration tools planned.

3.1.2 Design

In the Design phase, the system architecture and user experience components are created. This phase provides a visual and functional blueprint for development, aligning design elements with project requirements and user feedback.

- **System Architecture:** Develop use case, class, and sequence diagrams to visualize interactions and relationships among the system components, ensuring a maintainable structure for the application.
- **UI/UX Mockups:** Create wireframes and prototypes for key screens, such as the login page, weekly calendar view, task/event manager, and note-taking interface. Refine layouts and user flows to enhance usability and efficiency.
- **Database Schema Design:** Define and structure data tables and relationships to securely and efficiently store user profiles, tasks/events, and notes, along with backup and restoration capabilities.

- **Feature Interaction Design:** Outline task and event management workflows, note-taking mechanisms, and the layout of the weekly calendar interface to ensure a smooth and intuitive user experience.

3.1.3 Development & Integration

This phase focuses on constructing and integrating the application's frontend, backend, and database. Each component is carefully developed, tested for compatibility, and optimized for seamless performance.

- **Frontend Development:** Develop the user interface using modern web technologies such as HTML, CSS, JavaScript, and frameworks like React. Focus on creating an interactive and intuitive experience for managing tasks/events, and notes, with a user-friendly weekly calendar and task management system.
- **Backend & API Development:** Build server-side logic and develop RESTful APIs to handle core functionalities, including user authentication, task/event management, note handling, and secure data retrieval. Optimize backend processes to ensure robust and reliable application performance.
- **Database Integration:** Design and connect a structured database to store user profiles, tasks/events, and notes. Ensure secure data storage and implement backup and restoration capabilities to maintain data integrity and reliability.
- **Calendar & Task Management Integration:** Implement and integrate essential calendar and task management features. Synchronize the frontend and backend systems to provide seamless real-time updates and ensure consistent user experience across all devices.

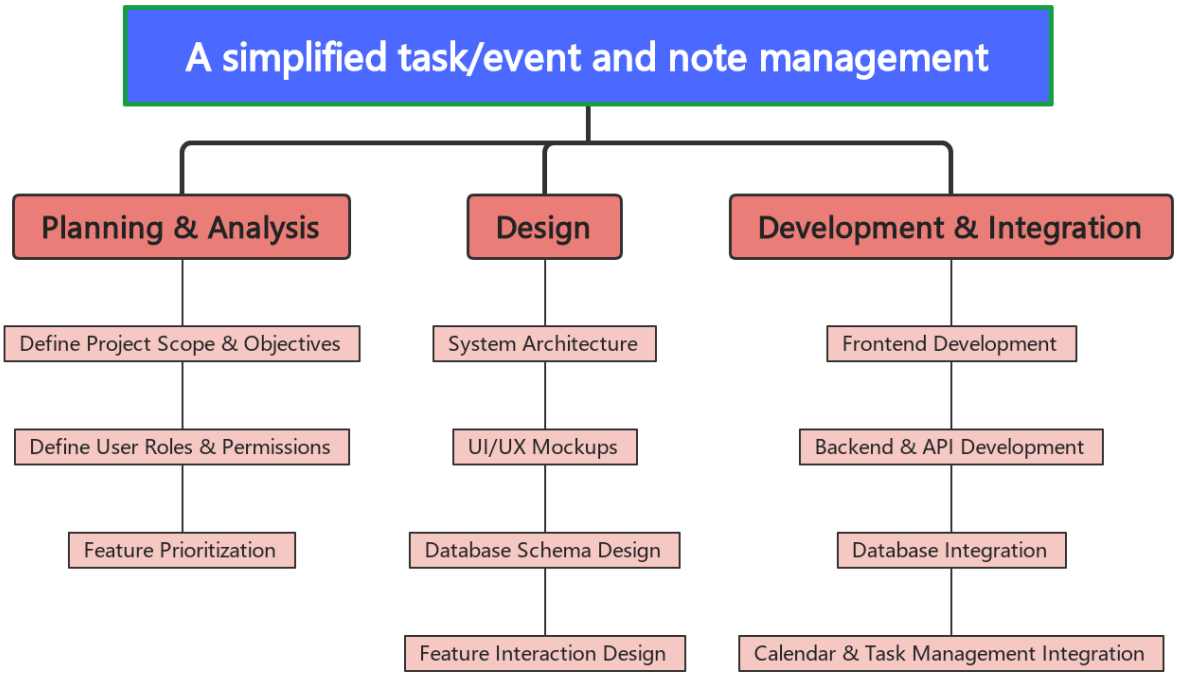

Figure 1: Work Breakdown Structure

3.2 Software Features

The "EasyDay" application is designed with a range of features to provide an organized, efficient, and user-friendly experience for task, event, and note management. These features are tailored to meet the needs of a diverse user base, from students to professionals. Below are the primary features of the system:

3.2.1 User Authentication

- **Login and Registration**: Users can securely register for an account and log in using their credentials. This ensures personalized access to their tasks/events, and notes.
- **Session Management**: Once logged in, users maintain a session, enabling seamless navigation between features without needing to reauthenticate.

3.2.2 Task and Event Management

- **Task Creation and Editing:** Users can create, edit, and delete tasks/events, including options for recurring tasks/events.
- **Crossed Out Finished tasks/events:** Users can mark tasks/events as done, automatically crossing them out on their weekly calendar, providing a visual cue of progress.

- **Event Scheduling:** A dynamic calendar interface allows users to add and manage events details, ensuring proper scheduling.

3.2.3 Note Management

- **Interactive Note System:** Users can create, edit, and delete notes efficiently.
- **Note Priority System:** Created notes will be sorted by deadlines, ensuring users do not miss their due dates.

3.2.4 Calendar Integration

- **Weekly View:** The vertical weekly calendar provides an intuitive view of tasks/events and events, enabling users to plan their week at a glance.
- **Real-Time Updates:** Changes to tasks/events are reflected in real-time, ensuring an up-to-date schedule.

3.3 System's Workflow

EasyDay is a task, project, and calendar management application designed to provide users with a streamlined workflow for organizing their schedules and responsibilities. The system workflow includes user authentication, task and project management, calendar navigation, and data persistence.

3.3.1 Login Page

- **Signup:**
  - New users can create an account by providing a unique username and password.
  - Once the signup is successful, the user is shown a success message and redirected to the login page for authentication.
- **Login:**
  - Users can log in by entering their username and password.
  - Upon successful login, users are redirected to the **main page**, where they can begin managing their tasks/events and notes.

3.3.2 Main Page

3.3.2.1 Monthly Calendar:

- The monthly calendar view shows the current month, with days categorized by weekdays and weekends.
- Users can easily navigate between months, ensuring a clear overview of their upcoming schedule.

3.3.2.2 Weekly Calendar:

- Users can view all tasks/events scheduled for the current week in a vertical calendar layout.

- The calendar allows users to navigate forward and backward to see tasks/events for previous or upcoming weeks.

3.3.2.3 Task Management

- **Create Task/Events:**
  - Authenticated users can create new tasks/events by providing essential details, such as task title, description, date, and time.
  - Tasks/events can also be set to repeat on a daily or weekly basis, allowing users to automate recurring tasks/events.
- **View tasks/events:**
  - Users can view their tasks/events organized by the current week in a vertical calendar format.
  - Tasks/events are categorized into morning, noon, and night slots for easy reference and planning.
- **Edit Task/Event:**
  - Users can open existing tasks/events to modify details, including title, description, date, or time.
  - The changes are saved and updated in real-time, reflecting immediately in the calendar view.
- **Delete Task/Event:**
  - Tasks/events can be deleted using the context menu or through the task modal, with a confirmation prompt to prevent accidental deletions.
- **Drag-and-Drop Functionality:**
  - Users can rearrange their tasks/events within the weekly calendar by simply dragging them to a new time slot or day.
  - When a task/event is dropped, its date and time are automatically updated, allowing for seamless adjustments to the user's schedule.

3.3.2.4 Note Management

- **Create Note:**
  - Authenticated users can create new notes by providing a project name, deadline, and a detailed description.
  - The note will be added to the user's note list and visible in the side panel.
- **View notes:**
  - Users can view all their notes, organized by deadline.
  - Notes are listed in the side panel for easy access and management.
- **Edit note:**
  - Users can open a note to edit its details and make updates such as changing the note name, description, or deadline.

    - Changes are saved instantly, keeping the note list up to date.
- **Delete note:**
    - Notes can be deleted by selecting the "Delete" option from the context menu.
    - A confirmation prompt ensures that the user intends to delete the note, preventing accidental removal.

3.3.2.5 Context Menu Actions

- **Open Context Menu:**
    - Users can right-click on tasks/events or notes to open a context menu with options such as opening, editing, or deleting the item.
- **Context Menu Actions:**
    - The context menu offers a quick and intuitive way to manage tasks/events and notes, making it easier to perform actions like editing or removing items without navigating away from the main interface.

3.3.3 Data Persistence

- **Local Storage:**
    - All user data, including tasks/events and notes, are stored securely in the browser's local storage.
    - This data is retrieved upon login and saved automatically when tasks/events or notes are created, updated, or deleted.
    - This ensures that user data remains persistent across sessions without the need for server-side storage.

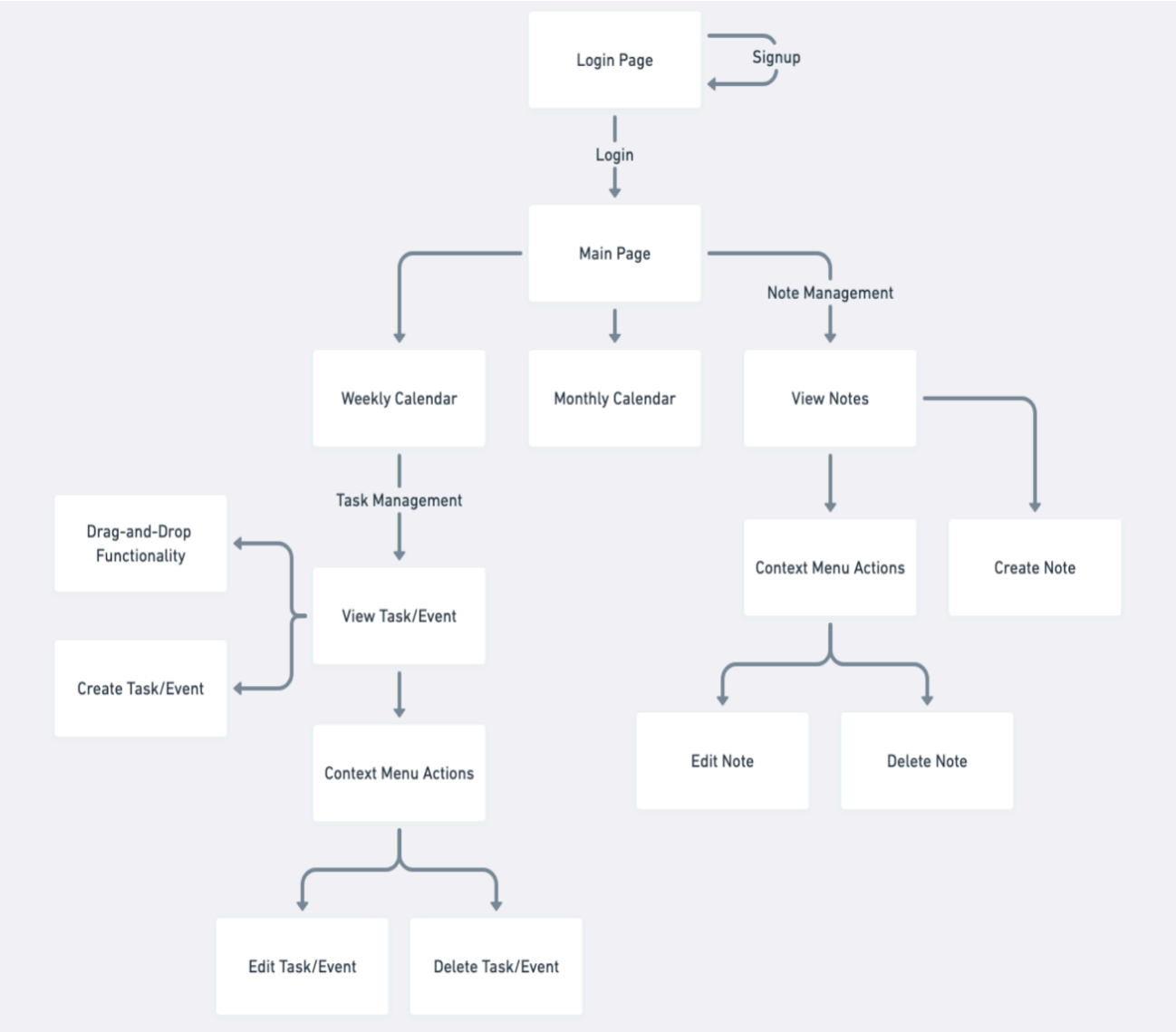

Figure 2: System Workflow

IV. Chapter 4: System Analysis and Design

4.1 Requirements Analysis

4.1.1 Functional Requirements

4.1.1.1 Authentication and Authorization

- **User Signup**: The system shall allow new users to create an account by providing a unique username and password.
- **User Login**: The system shall allow users to log in using their username and password.

- **Session Management**: The system shall maintain user sessions to keep users logged in across different pages.

4.1.1.2 Task Management

- **Create Task**: The system shall allow authenticated users to create tasks by providing a title, description, date, and time.
- **View Tasks**: The system shall allow users to view tasks for the selected week in a vertical calendar view.
- **Edit Task**: The system shall allow users to edit existing tasks.
- **Delete Task**: The system shall allow users to delete tasks.
- **Repeat Tasks**: The system shall allow users to create tasks that repeat daily or weekly.

4.1.1.3 Project Management

- **Create Project**: The system shall allow authenticated users to create projects by providing a name, deadline, and content.
- **View Projects**: The system shall allow users to view a list of their projects, sorted by deadline.
- **Edit Project**: The system shall allow users to edit existing projects.
- **Delete Project**: The system shall allow users to delete projects.

4.1.1.4 Calendar Views

- **Weekly Calendar**: The system shall display tasks for the current week and allow users to navigate to previous or next weeks.
- **Monthly Calendar**: The system shall display a monthly view with days categorized by weekdays and weekends, and allow users to navigate to previous or next months.

4.1.1.5 Data Persistence

- **Local Storage**: The system shall store user data, including tasks and projects, in the browser's local storage.

4.1.1.6 Drag and Drop

- **Task Drag and Drop**: The system shall allow users to drag and drop tasks to different time slots within the weekly calendar.

4.1.1.7 Context Menu

- **Open Context Menu**: The system shall allow users to right-click on tasks or projects to open a context menu with options to open or delete the item.
- **Context Menu Actions**: The system shall allow users to open the task/project modal or delete the task/project using the context menu.

4.1.2 Non-Functional Requirements

4.1.2.1 Performance

- **Response Time**: The system shall provide a response time of less than 2 seconds for all user interactions.
- **Scalability**: The system shall be able to handle up to 1000 concurrent users without performance degradation.

4.1.2.2 Usability

- **User Interface**: The system shall provide an intuitive and user-friendly interface for managing tasks, projects, and schedules.
- **Accessibility**: The system shall be accessible to users with disabilities, following WCAG 2.1 guidelines.

4.1.2.3 Reliability

- **Availability**: The system shall have an uptime of 99.9%, ensuring high availability for users.
- **Data Backup**: The system shall regularly back up user data to prevent data loss.

4.1.2.4 Maintainability

- **Code Quality**: The system shall follow best practices for code quality, including modular design and proper documentation.
- **Error Handling**: The system shall provide comprehensive error handling to ensure smooth operation and user experience.

4.2 Use Case Diagram

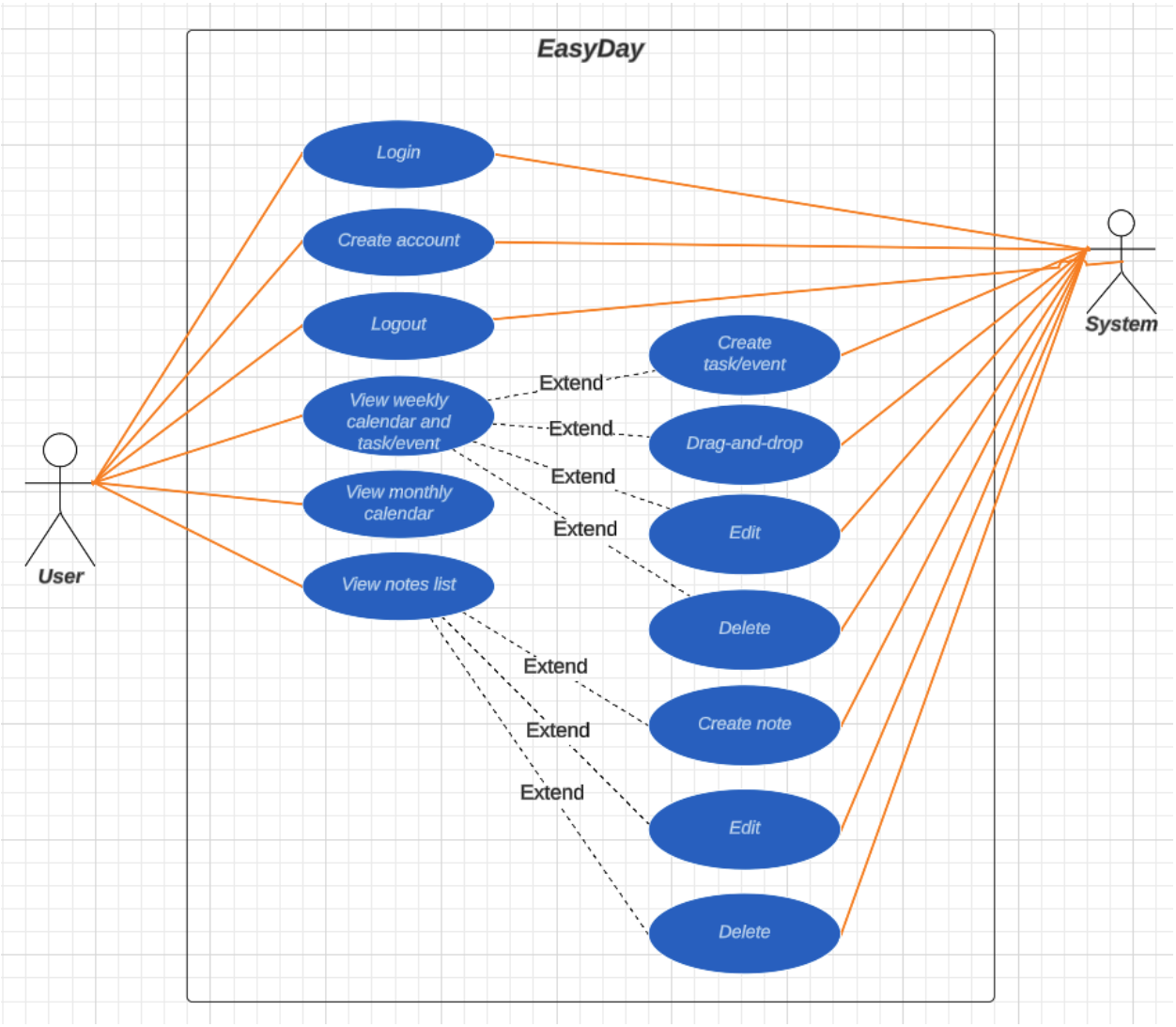

Figure 3: Use Case Overall System

4.2.1 User Story:

4.2.1.1 Login

- **Scenario 1: Successful Login**
  - As a user, I want to log in with valid credentials so that I can access my tasks and projects.
  - **Acceptance Criteria**:
    - The system verifies the username and password.
    - The user is redirected to the dashboard upon successful login.
- **Scenario 2: Unsuccessful Login**

    - As a user, I want to receive an error message if I enter incorrect login credentials so that I can retry with the correct information.
    - **Acceptance Criteria**:
        - The system shows an error message like “Invalid username or password.”
        - The login page remains accessible for retry.

4.2.1.2 Create Account

- **Scenario 1: Successful Account Creation**
    - As a new user, I want to create an account by providing valid information so that I can start managing my tasks.
    - **Acceptance Criteria**:
        - The system saves the user’s details and displays a success message.
        - The user is redirected to the login page or logged in automatically.
- **Scenario 2: Account Creation Fails**
    - As a new user, I want to be notified if I provide invalid or duplicate information during account creation so that I can fix the issue.
    - **Acceptance Criteria**:
        - The system shows specific error messages (e.g., "Username already taken" or "Invalid email format").

4.2.1.3 Logout

- **Scenario 1: Successful Logout**
    - As a user, I want to log out so that I can securely leave the application.
    - **Acceptance Criteria**:
        - The session ends, and the user is redirected to the login page.

4.2.2 Task/Event Management

4.2.2.1 Create Task/Event

- **Scenario 1: Successful Task/Event Creation**
    - As a user, I want to create a task/event by filling out a form with relevant details so that it appears in my schedule.
    - **Acceptance Criteria**:
        - The task/event is saved and displayed in the calendar or task list.
- **Scenario 2: Task/Event Creation Fails**
    - As a user, I want to see an error message if I forget to fill in required fields while creating a task/event so that I can fix it.
    - **Acceptance Criteria**:
        - The system highlights missing fields and shows a message like “Title is required.”

4.2.2.2 Drag-and-Drop Task/Event

- **Scenario 1: Successful Drag-and-Drop**
  - As a user, I want to move a task/event to a different time slot on the calendar using drag-and-drop functionality so that I can adjust my schedule easily.
  - **Acceptance Criteria**:
    - The task/event updates its date and time in the database.
    - The calendar view reflects the updated schedule.

4.2.2.3 Edit Task/Event

- **Scenario 1: Successful Edit**
  - As a user, I want to update the details of a task/event so that my schedule reflects the latest changes.
  - **Acceptance Criteria**:
    - The system saves the changes and updates the calendar view.
- **Scenario 2: Edit Fails**
  - As a user, I want to see an error message if the edit action fails (e.g., due to network issues) so that I can try again.
  - **Acceptance Criteria**:
    - The system retains the current data and displays an error message.

4.2.2.4 Delete Task/Event

- **Scenario 1: Successful Deletion**
  - As a user, I want to delete a task/event so that it no longer appears in my schedule.
  - **Acceptance Criteria**:
    - The task/event is removed from the calendar and database.
- **Scenario 2: Delete Confirmation**
  - As a user, I want to see a confirmation prompt before deleting a task/event so that I can avoid accidental deletions.
  - **Acceptance Criteria**:
    - The system displays a prompt: “Are you sure you want to delete this task?”

4.2.3 Notes Management

4.2.3.1 Create Note

- **Scenario 1: Successful Note Creation**
  - As a user, I want to create a note with a title and content so that I can save important details not tied to specific tasks.
  - **Acceptance Criteria**:

        - The note is saved and displayed in the notes list.
- **Scenario 2: Missing Fields**
    - As a user, I want to see an error message if I try to save a note without a title so that I know to add it.
    - **Acceptance Criteria**:
        - The system shows a message like “Title is required.”

4.2.3.2 Edit Note

- **Scenario 1: Successful Edit**
    - As a user, I want to update the content of an existing note so that it reflects accurate information.
    - **Acceptance Criteria**:
        - The system saves the changes and updates the note.
- **Scenario 2: Edit Fails**
    - As a user, I want to see an error message if my note edits cannot be saved so that I can retry.
    - **Acceptance Criteria**:
        - The system retains the old content and shows a message like “Unable to save changes. Please try again.”

4.2.3.3 Delete Note

- **Scenario 1: Successful Deletion**
    - As a user, I want to delete a note so that I can remove outdated or irrelevant information.
    - **Acceptance Criteria**:
        - The task/event is removed from the list and database.
- **Scenario 2: Delete Confirmation**
    - As a user, I want to see a confirmation prompt before deleting a note so that I can avoid accidental deletions.
    - **Acceptance Criteria**:
        - The system displays a prompt: “Are you sure you want to delete this note?”

4.3 Class Diagram

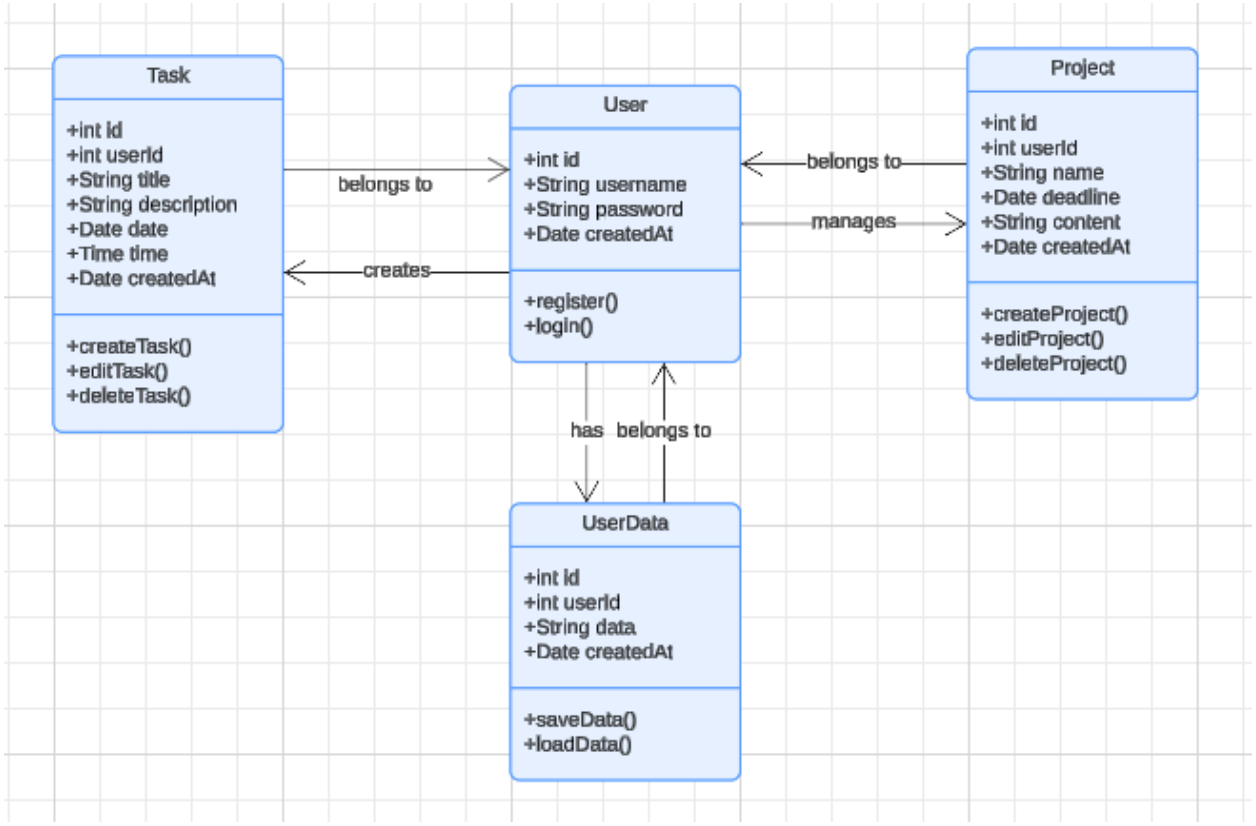

Figure 4: Class Diagram

4.4 Sequence Diagram

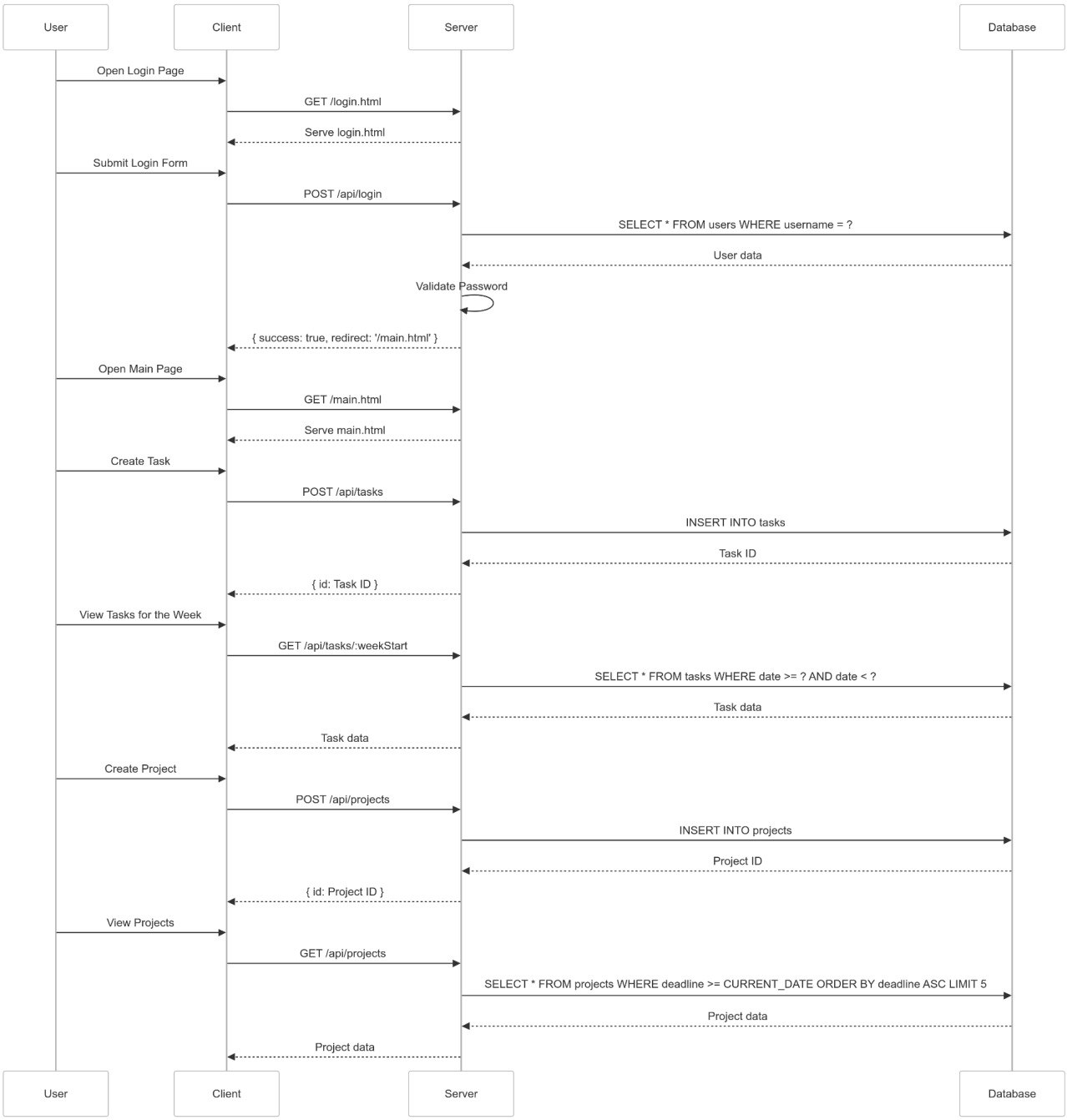

Figure 5: Sequence Diagram

V. Chapter 5: Backend System Architecture

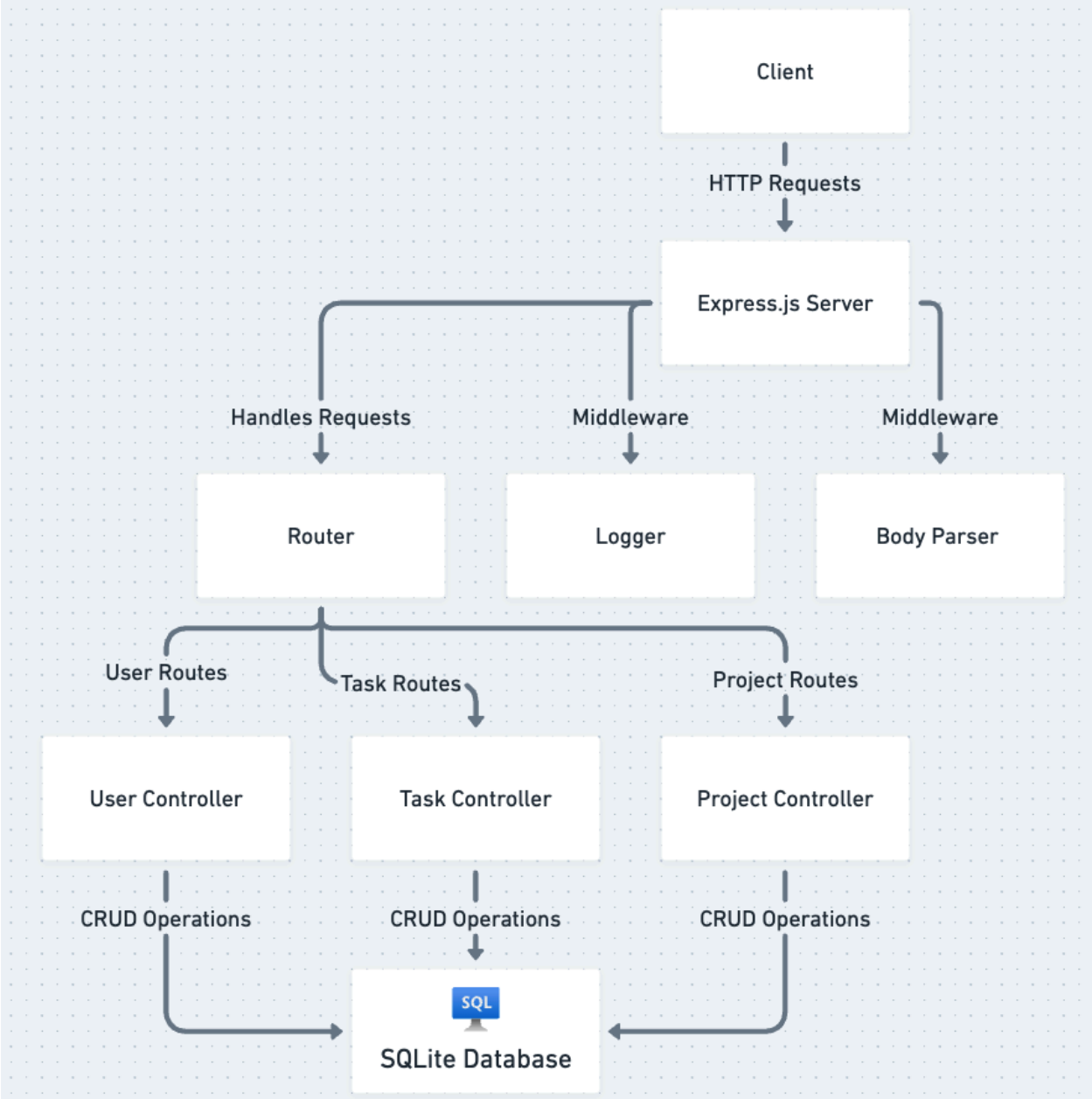

Figure 6: System Design for Backend

5.1 Components

5.1.1. Server

- **Express.js**: The primary server framework handles HTTP requests and responses, providing a lightweight and flexible foundation for the backend.

5.1.2. Database

- **SQLite**: The database system used for EasyDay. It stores structured information related to users, tasks, projects, and other essential data in a lightweight and portable manner.

5.1.3. Middleware

- **Logger**:
  - **Morgan** is implemented to log HTTP requests, aiding in debugging and monitoring the system.
- **Body Parser**:
  - Express.js's built-in body parser parses incoming JSON and URL-encoded payloads to enable seamless data exchange.
- **Cookie Parser**:
  - **Cookie-parser** parses cookies included in HTTP requests, supporting session management and user preferences.
- **Session Management**:
  - **Express-session** ensures users remain logged in across different pages during a session, handling session state securely.
- **Authentication**:
  - Custom middleware verifies user authentication, restricting access to protected resources and APIs.

5.2 Component Details

5.2.1 Express.js Server

- **Description**: Acts as the entry point for HTTP requests from the client, routing them to relevant controllers.
- **Responsibilities**:
  - Serve static files as needed.
  - Process API requests for various functionalities (tasks, user data, projects).
  - Integrate middleware to enhance request handling and security.

5.2.2 SQLite Database

- **Description**: Stores all critical application data with an emphasis on lightweight and local portability.
- **Tables**:
  - users: Contains user data, including:
    - ID
    - Username

        - Encrypted password
        - Account creation timestamp.
    - tasks: Stores task-related data:
        - Task ID
        - Associated user ID
        - Title
        - Description
        - Date and time
        - Task creation timestamp.
    - projects: Holds project information:
        - Project ID
        - User ID association
        - Project name
        - Deadline and content details
        - Creation timestamp.
    - user_data: Maintains additional user-specific information:
        - User ID
        - Data entries and metadata.

5.2.3 Middleware

5.2.3.1 Logger

- **Morgan** logs HTTP requests to provide insights into system behavior and identify potential issues.

5.2.3.2 Body Parser

- Built into **Express.js**, this middleware efficiently parses JSON and URL-encoded data, enabling robust data handling in the backend.

5.2.3.3 Cookie Parser

- Handles incoming cookies for user-specific preferences and authentication tokens.

5.2.3.4 Session Management

- **Express-session** ensures that authenticated users can maintain their session state while navigating the application.

5.2.3.5 Authentication Middleware

- Validates user credentials before granting access to sensitive routes and APIs, enhancing system security.

VI. Chapter 6: Minimum Viable Product

6.1 Login page:

- User is at the login page

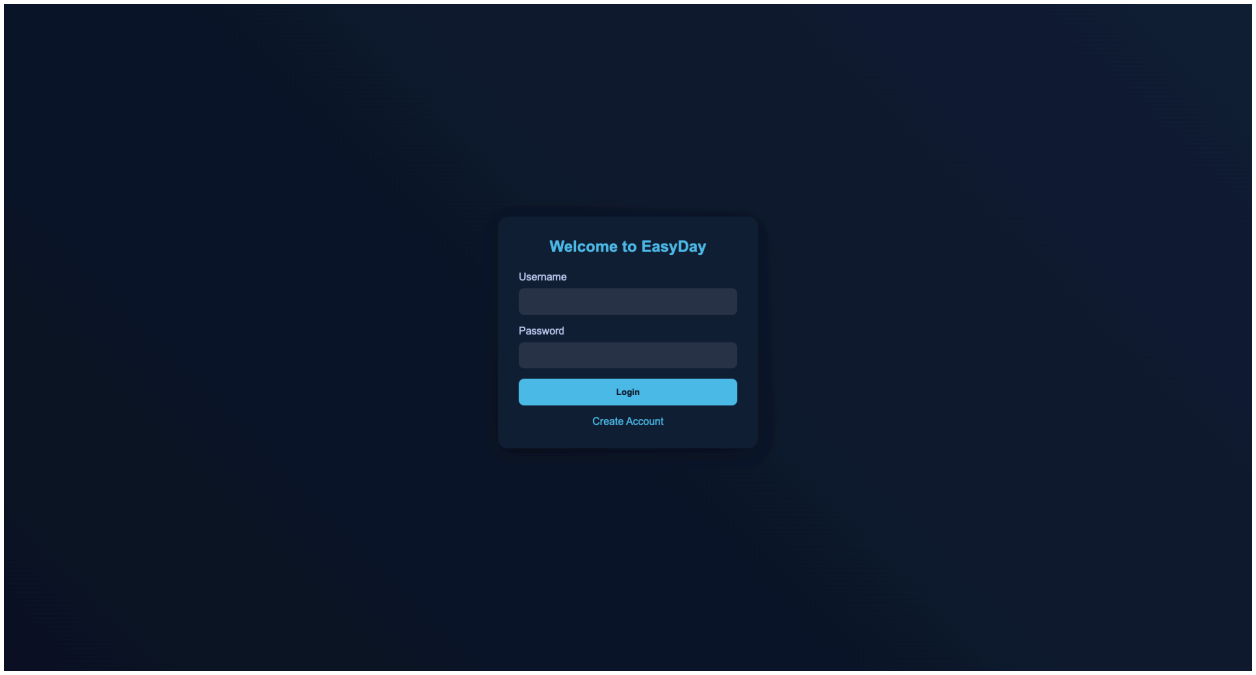

Figure 7: Login Page

- If user input incorrect username or password

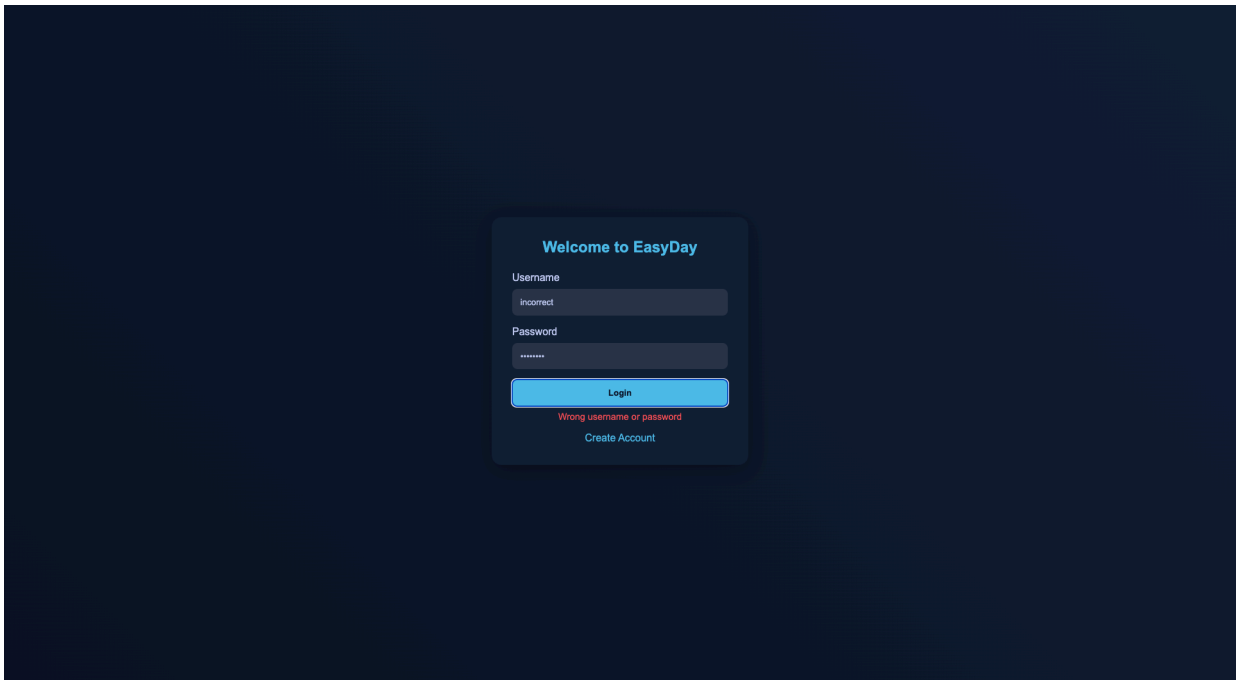

Figure 8: Demo Login

- If the User does not have an account, They can Create Account

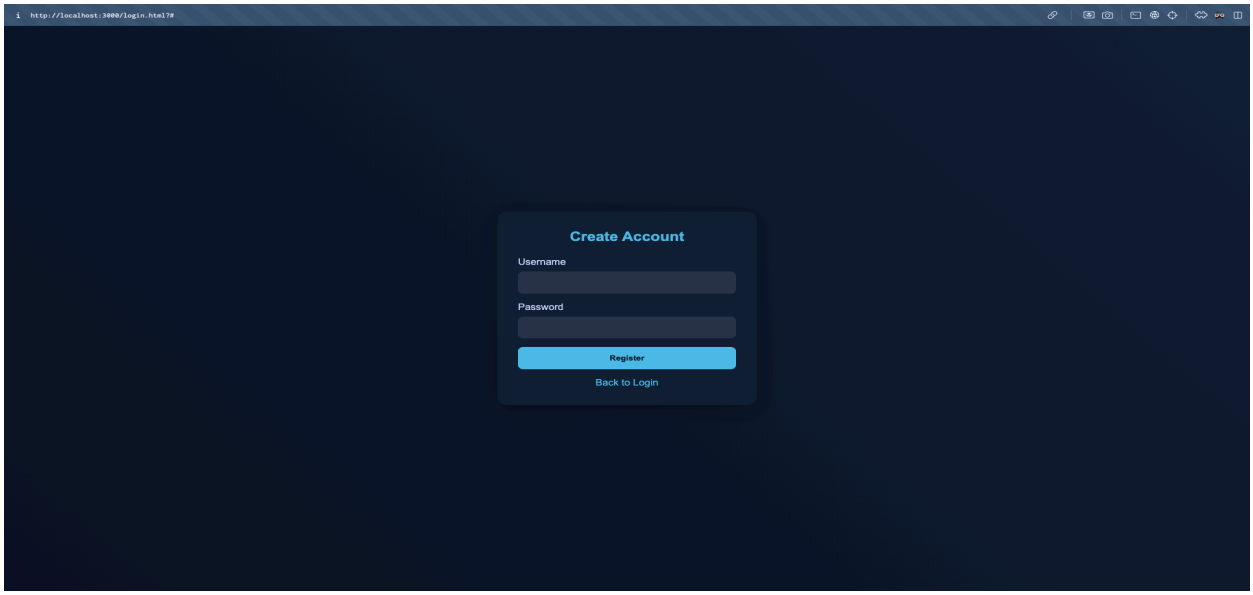

Figure 9: Demo Register

- Create account success / fail

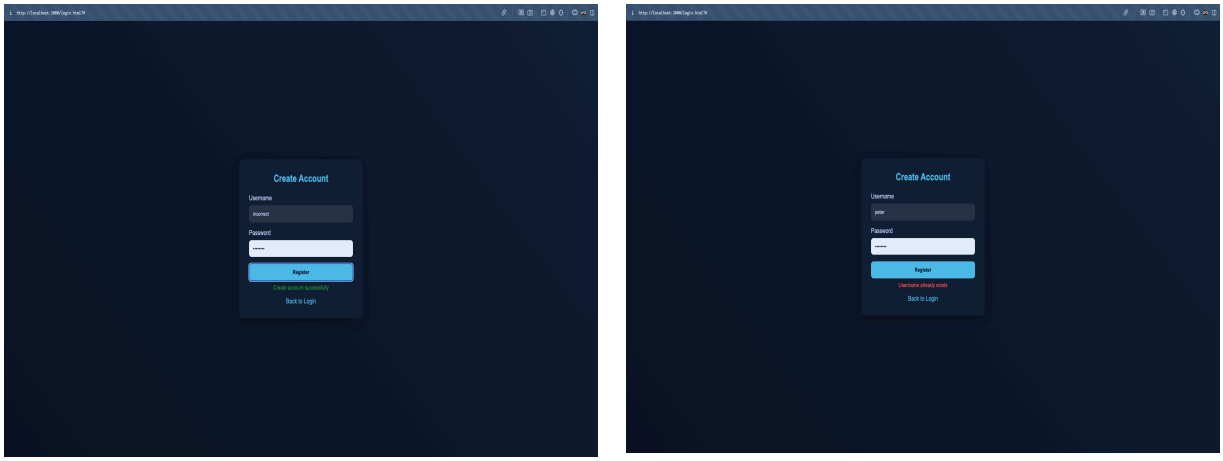

Figure 10: Demo Register - Failed

6.2 Main page

- After successfully Login, the system redirect user to Main Page with their personal data loaded

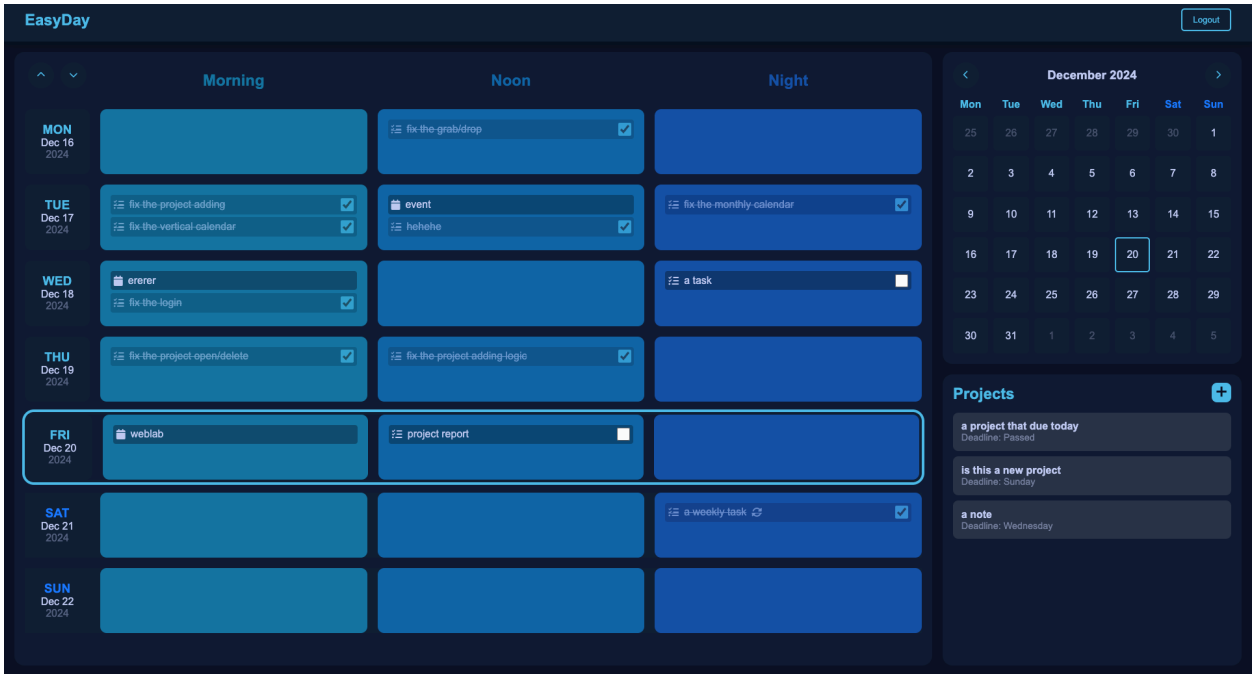

Figure 11: Main Page

- User create a new task/event and open an existing one (cannot change the Type and Repeat of an existed task/event)

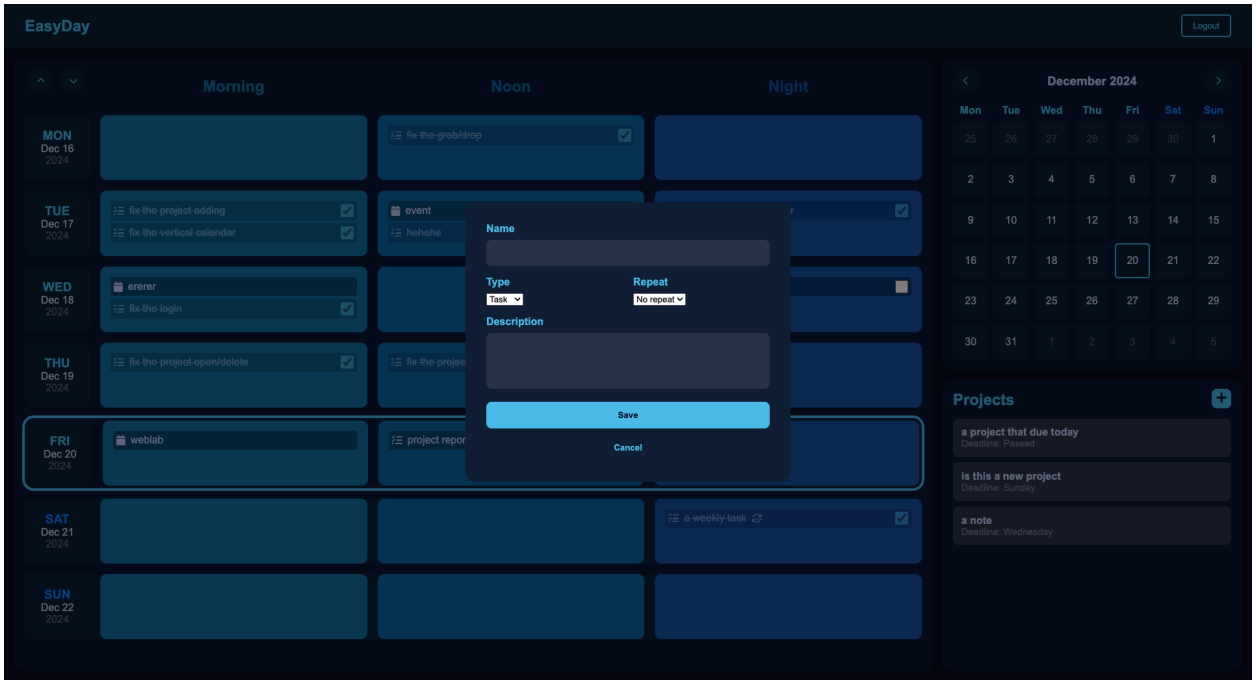

Figure 12: Create a new task/event

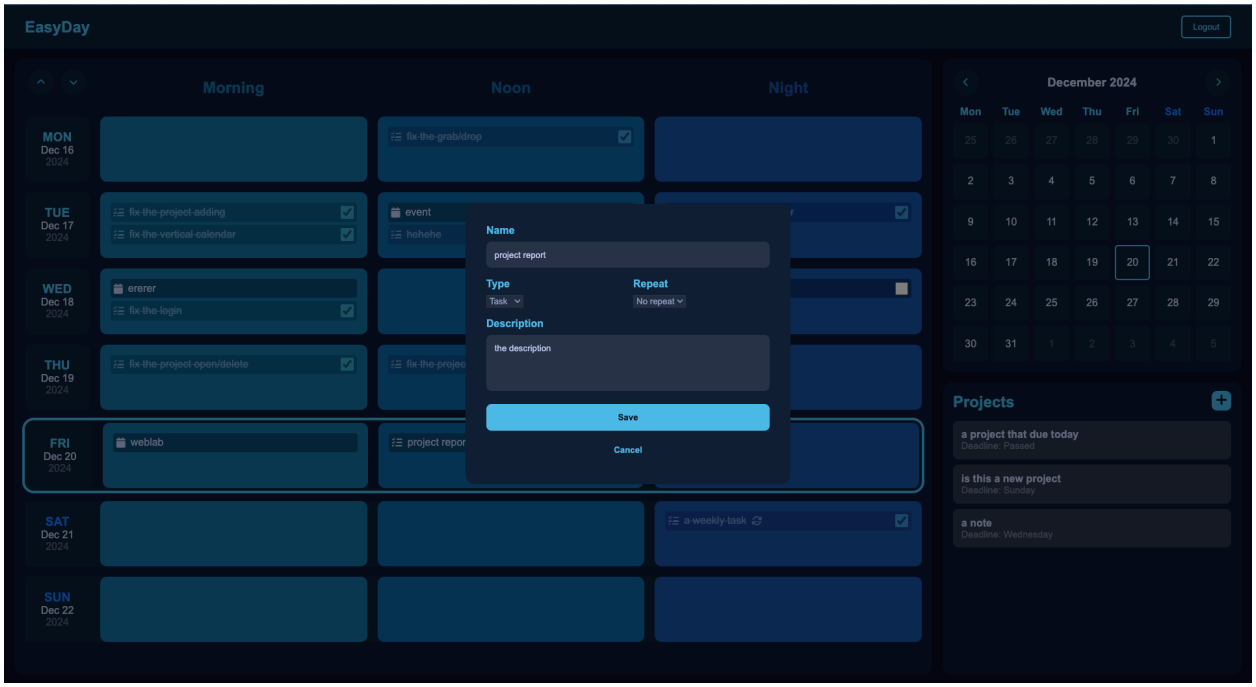

Figure 13: Open an existing task/event

- User Create a new Note (project)

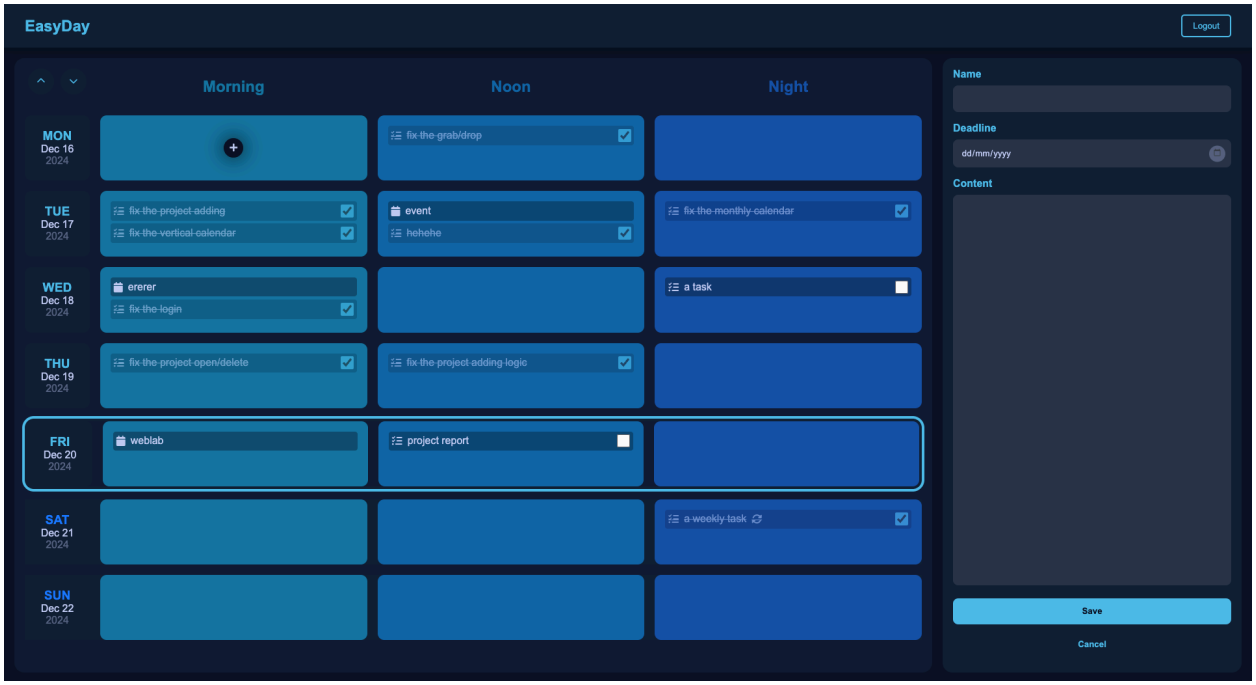

Figure 14: Create a new note

VII. Chapter 7: Achievements and Future Plan

7.1 Achievements

The development of "EasyDay" has accomplished several significant milestones, setting a strong foundation for task, event, and note management in a streamlined, user-friendly platform. Key achievements include:

- **Secure User Account Management**
  - Successfully implemented a secure login and registration system with encrypted user credential storage.
  - Developed basic profile features, enabling users to personalize their settings and tailor the platform to their preferences.
- **Core Features**
  - **Vertical Weekly Calendar Interface**: Designed and implemented a user-centric calendar interface to visually organize tasks and events by week, improving accessibility and productivity.
  - **Task/Event Management**: Provided robust functionality for users to add, repeat, edit, and delete tasks/events with ease.
  - **Note Management System**: Enabled users to create, edit, delete, and display notes, offering a comprehensive note-taking feature integrated with other task management tools.

- **Data Management**
    - Developed a secure backend system for storing user data, tasks/events, and notes in a structured database, ensuring quick access and seamless functionality.
    - Implemented a backup and restore feature to protect against data loss, ensuring the integrity of user information even in unexpected situations.
- **Streamlined User Experience**
    - Focused on delivering a clean, user-friendly interface that simplifies daily organization and offers intuitive interactions across all core features.

7.2 Future Plan

Despite the significant progress made, "EasyDay" has room for expansion and improvement. The following future plans outline the next steps for enhancing the platform and user experience:

- **Enhanced Note-Taking Interface**
    - Redesign the note-taking UI for a more intuitive and visually appealing experience.
    - Add features such as rich text formatting, categorization, and search functionality to improve note organization and retrieval.
- **Collaboration Features**
    - Introduce shared task/event and note management, enabling multiple users to collaborate on shared projects or schedules.
    - Provide role-based permissions for contributors, such as viewer or editor, to ensure effective collaboration and data security.
- **Mobile Application Development**
    - Develop a mobile version of "EasyDay" to enhance accessibility and allow users to manage tasks/events and notes on the go.
- **Improved Task and Event Customization**
    - Expand task/event options to include advanced scheduling features, such as priority levels, reminders, and tagging for better organization.
- **Advanced Data Analytics**
    - Incorporate analytics to provide users with insights into their productivity trends, task completion rates, and time management patterns.
- **Integration with Third-Party Tools**
    - Enable integration with popular productivity tools such as Google Calendar, Outlook, and task managers for seamless synchronization and expanded functionality.
- **Cloud Data Synchronization**

- Implement cloud-based data synchronization, allowing users to access their tasks, events, and notes across multiple devices in real-time.

VIII. Chapter 8: Conclusion

In conclusion, *EasyDay* represents a comprehensive and user-friendly web application designed to streamline task, event, and note management. By combining a vertical weekly calendar, a robust note-taking system, and an intuitive interface, the platform provides users with a single, cohesive solution for daily organization. Unlike conventional tools, *EasyDay* integrates key functionalities to enhance productivity and simplify the user experience.

The development process employed modern web technologies and a structured database system, ensuring secure data management and scalability. By utilizing the Scrum methodology, the project team maintained a focus on iterative improvements and user feedback, resulting in a polished and practical application.

EasyDay is positioned to meet the needs of students, professionals, and anyone seeking an efficient organization tool. The planned future enhancements, including reminders, customization options, and collaboration tools, will further expand the application's capabilities, making it even more versatile and valuable to its users.

With its strong foundation and commitment to continuous improvement, *EasyDay* stands as a reliable and innovative productivity platform, ready to adapt to the evolving needs of its audience.

IX. Chapter 9: References

- GitHub Link: https://github.com/PeterNguyenX/EasyDay