Infosys SpringBoard Virtual Internship 6.O Documentation

Name: Muppidi Sri Kumar

Email: sreekumarmuppidi@gmail.com

Module-1:

Date: 01/12/2025 – 03/12/2025

Task: Create project structure

1. Objective:-
To create the basic project structure for the Task Reminder application using Spring Boot.

2. Steps Taken:-

- Created a new Spring Boot project using Maven.
- Created required packages for controller, model, repository, and config.
- Set up folders for templates and static resources.
- Verified that the project runs successfully.

3. Challenges Encountered:- N/A

4. Verification:-

- Successfully ran the application and confirmed that the project structure is created properly.

Date: 04/12/2025 – 05/12/2025

Task: Create HTML page and write code for tables to store data

1. Objective:-
To design the initial UI and create tables to display task data.

2. Steps Taken:-

- Created HTML page using Thymeleaf.
- Designed a table to show task details.
- Linked CSS for basic styling.
- Integrated the page with controller.

3. Challenges Encountered:-
Faced some alignment issues while designing the table.

4. Verification:-

- By displaying sample data in the table, verified that the UI is working properly.

Date: 08/12/2025 – 10/12/2025

Task: Update controller, add view and edit buttons

1. Objective:-
To implement view and edit functionalities for tasks.

2. Steps Taken:-

- Updated controller methods for handling view and edit requests.
- Added View button to display task details.

- Added Edit button to modify task information.
- Tested navigation between pages.

3. Challenges Encountered:-
Faced Whitelabel error due to incorrect URL mappings.

4. Verification:-
- By clicking on view and edit buttons and getting correct pages, verified the functionality.

Date: 11/12/2025 – 12/12/2025

Task: SQL integration

1. Objective:-
To integrate SQL database with the application for storing tasks.

2. Steps Taken:-
- Installed and configured MySQL database.
- Created database schema.
- Added datasource properties in application.properties.
- Integrated JPA for CRUD operations.
- Tested storing and fetching data from database.

3. Challenges Encountered:-
Faced database connection issues initially.

4. Verification:-
- By adding tasks and verifying entries in the database, confirmed successful integration

---

Date: 15/12/2025

Task: Module 2 begins – Styling for project

1. Objective:-
To improve the UI of the application using CSS.

2. Steps Taken:-

- Styled forms, tables, and buttons using CSS.
- Improved layout and alignment of pages.
- Added dark mode feature.
- Enhanced overall look and feel of UI.

3. Challenges Encountered :-
Adjusting layout for different pages took some time.

4. Verification:-

- By checking UI changes in browser, verified improved appearance.

Date: 18/12/2025

Task: Actions, priority, status implementation

1. Objective:-
To implement task actions and categorization using priority and status.

2. Steps Taken:-

- Added delete and update actions for tasks.
- Implemented priority levels (Low, Medium, High).
- Implemented status options (Pending, In Progress, Done).
- Updated UI to reflect status and priority.

3. Challenges Encountered:-

Handling status updates in controller logic.

4. Verification:-

- By updating status and priority and seeing changes in UI, verified functionality.

Date: 22/12/2025

Task: JUnit testing

1. Objective:-

To learn and implement basic unit testing using JUnit.

2. Steps Taken:-

- Learned basics of JUnit framework.
- Wrote test cases for repository and controller methods.
- Executed tests and analyzed results.
- Fixed minor issues found during testing.

3. Challenges Encountered (if any):-

Understanding annotations and test setup initially.

4. Verification:-

- By running test cases successfully, verified correctness of the code.

Date: 23/12/2025

Task: Pagination, Card View, and Calendar View

1. Objective:-

To enhance the application by adding pagination and multiple views for better visualization of tasks.

2. Steps Taken:-

- Implemented pagination to display tasks page by page.
- Added Card View to show tasks in a card-based layout.
- Created Calendar View to display tasks based on due dates.
- Updated controller and UI to support switching between views.
- Tested navigation between different views.

3. Challenges Encountered :-

Faced issues in routing and view integration initially.

4. Verification:-

- By navigating through pages and switching between card and calendar views, verified that all features are working properly.

Date: 05/01/2026 – 08/01/2026

Task: User Authentication Implementation

1. Objective:

The objective of this task was to implement user authentication to secure the application and allow access only to registered users.

2. Steps Taken:

- Designed user registration and login pages.
- Created User model and repository.
- Implemented authentication controller for login and registration.
- Validated user credentials during login.
- Prevented multiple accounts with the same email or phone number.
- Tested login and logout functionality.

3. Challenges Encountered:

- Handling authentication errors and redirection issues.
- Resolving Whitelabel errors during login flow.

4. Verification:

- Verified that only registered users could log in successfully.
- Confirmed that duplicate user registration was restricted.

Date: 09/01/2026 – 10/01/2026

Task: Session Management Integration

1. Objective:

To maintain user login state across the application using session management.

2. Steps Taken:

- Implemented HttpSession to store logged-in user details.
- Added session checks for protected pages.
- Redirected unauthenticated users to login page.
- Implemented logout functionality by invalidating session.
- Ensured root URL redirects based on session state.

3. Challenges Encountered:

- Managing session scope across multiple controllers.
- Preventing direct access to secured URLs.

4. Verification:

- Verified session creation after login.
- Confirmed session invalidation on logout.
- Tested restricted page access without login.

Date: 11/01/2026 – 12/01/2026

Task: User-Specific Task Integration

1. Objective:

To ensure that each user has a separate task space where only their own tasks are visible.

2. Steps Taken:

- Established a relationship between User and Task entities.
- Modified task creation logic to associate tasks with the logged-in user.
- Updated task listing to fetch tasks based on session user.
- Applied user-specific logic across table view, card view, calendar view, and filter page.
- Ensured data isolation between different users.

3. Challenges Encountered:

- Fixing repository and import errors during user-task mapping.
- Ensuring consistent behavior across multiple task views.

4. Verification:

- Verified that each user could see only their own tasks.
- Confirmed new users received a fresh task page with no previous data.

Date: 13-Jan-2026 to 14-jan-2026

Task: Implementing Profile Page Feature

1. Objective:-

The objective of this task was to create a user profile page where each logged-in user can view their personal details and manage their account information.

2. Steps Taken:-

- Created a separate profile page to display user details such as name and email.
- Integrated session management to ensure the profile is user-specific.
- Implemented backend controller methods to fetch logged-in user data.
- Displayed user details dynamically using Thymeleaf.
- Added navigation options to return back to the home page.

3. Challenges Encountered (if any):-

- Handling session data correctly to show the profile of the logged-in user only.
- Resolving mapping conflicts and data binding issues.

4. Verification:-

- Verified that each user can see only their own profile details.

Date: 14-Jan-2026 / 16-jan-2026

Task: Implementing Dashboard Feature

1. Objective:-

The objective of this task was to design and implement a dashboard page for the Task Reminder application that provides users with a quick overview of their task statistics in a single place.

2. Steps Taken:-

- Designed a separate dashboard page (dashboard.html) for better visualization.
- Created a dedicated controller method to handle dashboard requests.
- Fetched user-specific task data from the database using session information.
- Calculated task statistics such as:
  - Total number of tasks
  - Completed tasks
  - Pending tasks
  - Overdue tasks
  - High-priority tasks
- Displayed these metrics using dashboard cards for easy understanding.

- Ensured the dashboard data is unique for each logged-in user.

3. Challenges Encountered:-

- Handling user-specific data filtering for dashboard metrics.
- Correctly calculating overdue and high-priority tasks based on dates and priority levels.

4. Verification:-

- Verified dashboard values by comparing them with actual task data in the database.
- Confirmed that different users see different dashboard statistics based on their own tasks

Date: 19/01/2026- 21/01/2026

Task: Adding Download Button for Excel Sheet of Tasks

1. Objective:-

The objective of this task was to allow users to download their task data in Excel format for offline access and reporting purposes.

2. Steps Taken:-

- Added a download button in the application UI for exporting tasks.
- Implemented backend logic to fetch user-specific task data from the database.

- Used Apache POI library to generate an Excel file dynamically.
- Populated the Excel sheet with task details such as task name, description, due date, priority, and status.
- Configured the controller to return the Excel file as a downloadable response.

3. Challenges Encountered (if any):-

- Handling file generation and response streaming correctly.
- Ensuring that only the logged-in user's tasks are included in the downloaded file.

4. Verification:-

- Successfully downloaded the Excel file and verified task data accuracy.
- Tested with multiple users to ensure user-specific task data is exported correctly.

Date: 21-Jan-26 to 23-Jan-26

Task: Email Reminder when a New Task is Added

Objective:
To notify users through email whenever a new task is created, ensuring they receive instant confirmation and awareness of task creation.

Steps Taken:

- Integrated email service with the task creation module.
- Triggered an automated email notification whenever a user adds a new task.
- Included task details such as task name and due date in the email content.

Challenges Encountered:

- Faced connection issues with SMTP configuration initially.

Verification:

- Successfully verified email delivery by adding new tasks and checking inbox notifications.

Date: 24-Jan-26 to 25-Jan-26

Task: Email Reminder for Tasks Near Due Date

Objective:
To remind users via email when a task's due date is approaching, helping them avoid missing deadlines.

Steps Taken:

- Implemented logic to identify tasks nearing their due date.
- Configured email reminders to be sent before the task deadline.
- Ensured reminders are sent only for pending tasks.

Challenges Encountered:

- Managing date comparisons and scheduling email triggers accurately.

Verification:

- Tested by setting near due dates and confirming reminder emails were received.

Date: 26-Jan-26 to 27-Jan-26

Task: Adding MIT License to the Project

Objective:
To apply an open-source license that allows free usage and distribution of the project while protecting the author legally.

Steps Taken:

- Added an MIT License file to the project root directory.
- Updated the README file to mention the project license.
- Pushed license changes to the GitHub repository.

Challenges Encountered:

- Understanding license selection and its implications.

Verification:

- Confirmed that the MIT License is visible on the GitHub repository.