SOFTWARE ENGINEERING

CSE3001

J Component Report

Title:

Frello -Your Friendly Project Management Tool

SUBMITTED BY: -

19BCE0133 Shruti Shreya

19BCE0235 Yash Ray

19BCE0365 Sourav Pandey

Name of faculty:

Prof SWATHI JN

(SCHOOL OF COMPUTER SCIENCE AND ENGINEERING)

Acknowledgement

We would like to express our gratitude to our Professor Dr. Swathi J. N. for her expert advice and encouragement throughout this project. Her guidance helped us complete our project on time. We specially thank her for co-operating with our doubts and questions and clarifying every little detail which was required. We would also like to thank our institute, Vellore Institute of Technology, Vellore, for giving us this opportunity for building a J Component for the course Software Engineering.

Table of Contents

List of Figures

1. Introduction

1.1 Objective

Projects are vast and time taking and organizing them is an ordeal. Our objective with Frello is to provide a friendly project management tool for students as well as professionals. With Frello, one can organise and schedule their tasks neatly while maintaining an interface suited to their projects. This will not only save the users a lot of confusion and frustration but will also clear up their personal life schedules which will positively affect their work efficiency.

1.2 Motivation

Whether one is a student or a working professional, one can find organising and maintaining project related tasks not only confusing and frustrating but also a bit cumbersome. While tasks are divided between people by project managers, a majority of working individuals feel the need for a proper management and organisation system even for their own tasks.

1.3 Background

The importance of project management can't be overstated. When it's done right, it helps every part of the project run more smoothly. It allows one’s team to focus on the work that matters, free from the distractions caused by tasks going off track or budgets spinning out of control. Project management is important not only among team members but also for every individual member’s tasks.

2. Project Description and Goals

2.1 Abstract

Our Project is a software which will help set personal goals for an individual. Developing a software requires determination apart from all other aspects and skills and what better way to stay determined than to set your own goals and meet them. It will enable every user to create an account and organize their work as per their own needs and comfort. They will be secured, so only that a particular user will be able to access and edit their timeline. Our Project is general-purpose, Web-based software, which can be used for complex multi-user projects for business. In addition to tracking the status of tasks, it includes a powerful system for logging and reporting time spent on tasks. Combined with powerful reporting mechanisms, it also provides many customisation options for a better view and interface.

2.2 Scope

To enable any user or organization to increase their daily efficiency by storing and maintaining different timelines and progress of their work routine or project deadlines privately/publicly as per their own needs and comfort and analyse their timetable.

2.3 Project Description

To create a platform for users/organizations to organize their work and store timeline of their work routine. Enable every user to create an account and

organize their work as per their own needs and comfort. They will be secured, so only that particular user/organization will be able to access and edit their timeline. They include work hours trackers, communication platforms, project and task organizers, scheduling software, team management systems, calendar apps, and report making tools, and many others. It also provides a customisation window where user can change background or text colour for better interactivity and interface. Lists created by users can store any project milestone or activities followed by the due deadline of their project. It can also store start and finish date for each project segment. Later all the details can be stored and viewed by the user in offline mode for better accessibility and user satisfaction.

2.4 Project Stakeholders

This Project may affect, be affected by, or perceive itself to be affected by a decision, activity, or outcome of a project.

1. Internal stakeholders
    - Project Leader
    - Project Members
        - →Backend Team
        - →Frontend Team

2. External stakeholders
    - Any customer interested in using our tool for their projects.
        - →Individual
        - →Organisation
    - Employees of the organisation undertaking a project using our tool.

3. Technical Specifications

3.0 Process Model

We will be following the **EVOLUTIONARY DEVELOPMENT** model.

The Evolutionary Development model focuses on Effective ways to meet immediate customer needs and Incremental Specification development. In our project, we will integrate a React based frontend, designed to help the user manage their project tasks and deadlines efficiently with a smooth running backend for enhanced user experience.

Our project focuses on customer's needs and seeks constant validation, feedback and refinement. Thus, the Evolutionary development model justifies the stipulation of the project.

Other Models fail to achieve the required prerequisite of the project.

Waterfall Model :

The waterfall model requires considerable initial commitments and restricts the user-based needs. Further, it is largely inflexible and will require significant re-work for even minute adjustments. As we plan on taking frequent feedback from users(as well as project reviews during the semester), repeated planning will slow the project down and will affect its quality. Therefore, waterfall model is not suitable.

Incremental Development Model:

Incremental development model is generally used for heavy projects to kick start the project and then deliver it part by part in increments. Our project is a straightforward project management tool with limited features for added user-friendliness, therefore more suitable to be delivered wholly at once. Incremental development will put unnecessary activities in transition, which would ultimately extend the stipulated time for project completion. Furthermore, all resources needed for our project such as open source software development tools, frontend and backend development software as well as designing tools are available at once, so there is no need to divide the project in increments.

Rapid Application Development Model:

Rapid application development model works for projects with short developmental cycles. Our project team is limited to 3 members and hence, it would be difficult to conduct our work parallelly among different team members, as advocated by the rapid application development model. Further, it could potentially increase the risk involved in our project and might cause a delay.

Spiral Model:

The Spiral model is fit for use in scenarios where a new technology is being used or a new area of development is being explored. Our project will use standard frontend and backend web development frameworks and all of the resources required are already available. Therefore, we see no requirement for the Spiral Model in our project.

3.1 Product Functions

The follow is a table of the requirements that the system SHALL meet.

Table of Requirements

ID	Stakeholder	Requirement
1	User	The System shall allow the user to create an account using their Google account.
2	*User*	The System shall allow the user to create multiple lists.
3	*User*	The System shall allow the user to create multiple tasks.
4	*User*	The System shall allow the user to create multiple sub-tasks.
5	*User*	The system shall allow the user to edit the task name.
6	*User*	The system shall allow the user to edit the sub-task name
7	*User*	The system shall allow the user to reorder the lists(Tasks) based on their priorities.
8	*User*	The System shall allow the user to reposition the cards(subtasks).

9	*User*	The System shall allow the user to delete Multiple Tasks.
10	*User*	The System shall allow the user to delete Multiple sub-Tasks.
11	*User*	The System shall allow the user to schedule tasks and deadlines.
12	*User*	The System shall allow the user to customize the scene background using pictures of his/her choice from a bunch of options available.
13	*User*	The System shall keep a record of the work done by the user for future uses.
14	*User*	The system shall allow the user to create an offline copy of the list created.
15	*User*	The System shall allow the user to logout.

3.2 Constraints

The following is a table of the design constraints that the system shall meet.

Table of Design Constraints

ID	Stakeholder	Requirement
1	User	The user shall only be able to set his/her personal goals and deadlines
2	Team members	The team members shall not view the user's individual progress or task distribution
3	User	The user shall provide his/her google account to facilitate authentication and login.
4	User	The user shall have a proper functioning browser

3.3 Entity Relationships

Figure 1 shows the entity relationships for the Task scheduler

Figure 1, Entity Relationship Diagram

3.4 Data Flows

The following figures represent the data flow diagrams of the Task scheduler.

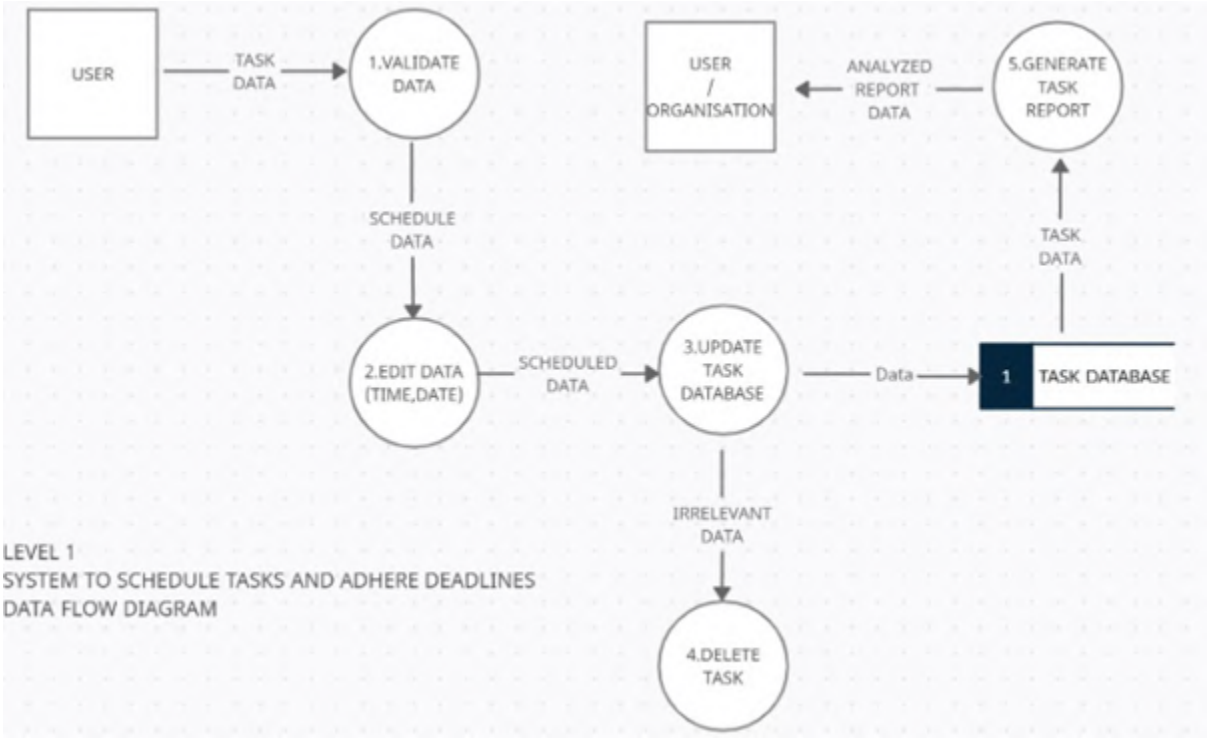


Figure 2, Data Flow Diagram

3.5 Specific Requirements

3.5.1 System Features

3.5.1.1 Signup / Login

Introduction

The Frello shall allow a user to create their account by signing up using email or google authentication.

Functional Requirements

Purpose: Creating a account for the user.

Input:. Email id, Name , age , Gender, Organization and other necessary details required for signing up.

Processing: Storing the user data in the database and verification of email id if needed.

Output: Account creation.

Stimulus Response

A) User creates his/her account.

User Actions	**System Actions**
(1) Get Started	
(2) Initiate signup window	
	(3) Close the home page
	(4) open up the signup page
(5) Fills the required inputs	
	(6) System Retrieves Information from the input form.

	(7) Stores in Database
	(8) Email Verfication
(10) Verifies the email	
	(11) Sign Up Window closes
	(12) Logs into user account

B) User logs into his/her account

User Actions	System Actions
(1) Get started	
(2) Initiate the Login Page	
	(3) Close the Home page
	(4) Open the login Window
(5) Fills email id and password	
(6) Logs In	
	(7) Verifies the email id and password
	(8) Closes the login Window
	(9) Logs into his/her account

3.5.1.2 Task Scheduling

The software schedules tasks based on user inputs and requirements.

Functional Requirements

Purpose: User Input/Initiating Task Scheduler.

Input: Project Name, Activities, Milestones, Deliverables, Deadlines, Start Date, End Date and other Project components.

Processing: Creates Lists.

Output: Task that are to be scheduled in a List.

Stimulus Response

User Actions	System Actions
(1) Clicks on New List Button	
	(2) Open the Input Dialog Box
(3) Gives Project Name as Input	
	(4) Creates a List with project name assigned by the user.
	(5) Add, Edit and Delete Buttons Enabled
	(6) New List Button Enabled
(7) User Input about Project Components	
	(8) Stores in Task Database
	(9) Gives output

3.5.1.3 Editing an Existing List

Introduction

This feature allows the user to edit his/her created List (Tasks that are to be scheduled or are past deadline)

Functional Requirements

Purpose: Editing an existing List.

Input: Project Name, Activities, Milestones, Deliverables, Deadlines, Start Date, End Date and other Project components.

Processing: New edited List.

Output: Task that are to be scheduled in a List.

Stimulus Response

A) Editing a Task

User Actions	System Actions
(1) Get started	

(2) Log In	
	(3) Opens the User home page
(4) Chooses the List to be edited	
(5) User Edits the List	
.	(6) Edited List stored in Database
	(7) Displays the Edited List
(8) User Deletes the List	
	(9) Deletes the List from Task Database
	(10) Displays the List

3.5.1.4 Customizing Background

Introduction

This feature allows the user to change the background to any picture or solid color of his choice which will be out of a set of options pre-available.

Functional Requirements

Purpose: Changing the Background view..

Input: Selected image

Processing: Background changes for that particular user.

Output: The View for that particular user gets updated..

Stimulus Response

User Actions	**System Actions**
(1) Click on the change background button	
	(2) Open a side pane with multiple options.
(3) Select a preferred background.	
	(4) The view gets updated to the chosen image or solid color.
(5) Clicks on close button	
	(6) The side pane closes

4. Design Approach and Details

4.1 Design Approach/Materials and Methods

4.1.1 Decomposition Description

4.1.1.1 Module Decomposition

The Frello Project Management Software has been decomposed into the following:

1. Authentication System Module : This module allows the user to sign in into his/her frello account using their Google account.

2. Task CRUD : This module allows the user to add tasks to their dashboard. These tasks can be modified, in the sense that new tasks can be created, tasks can be scheduled and displayed, updated and deleted. They can also be reordered.

3. Sub-Task CRUD : This module allows the user to add sub-tasks to specific tasks in their project. The same operations can be done on subtasks as on the tasks.

4. Customization: This module allows the user to change the background of their dashboard according to their own liking. A default set of backgrounds will also be provided to the user to choose from, which will include pictures as well as solid colours.

4.1.1.2 Data Decomposition

The following are the two major data components, the User Database and the Task Database

User Database: This is a database that contains the Login information of all the users including the

- username(email)
- password.

Task Database: This is a database that stores the details of the tasks and subtasks that the user adds to their projects. This information includes the

- Name of the tasks
- Task descriptions
- Deadlines of the tasks
- Subtasks under the particular task
  Properties of subtasks

4.1.2 Dependency Description

4.1.2.1 Inter-module Dependencies

Independent Modules

The following modules are independent and do not rely on any other modules to initiate them or to provide data.

- Authentication System Module

Dependent Modules

The following modules are dependent on one another for their functioning.

· Task CRUD Module: This module depends on the Authentication system module for access. It cannot be accessed or initiated unless the user is verified and logged in.
· Subtask CRUD: This module also depends on the Authentication system module for access. It cannot be accessed or initiated unless the user is verified and logged in. Subtask CRUD also depends on the Task CRUD module as there cannot be any subtasks without tasks.
· Customisation: This module depends on the Authentication system module for access. It cannot be accessed or initiated unless the user is verified and logged in.

4.1.3 Module Interface Description

4.1.3.1 Authentication System Module

User Interface Design

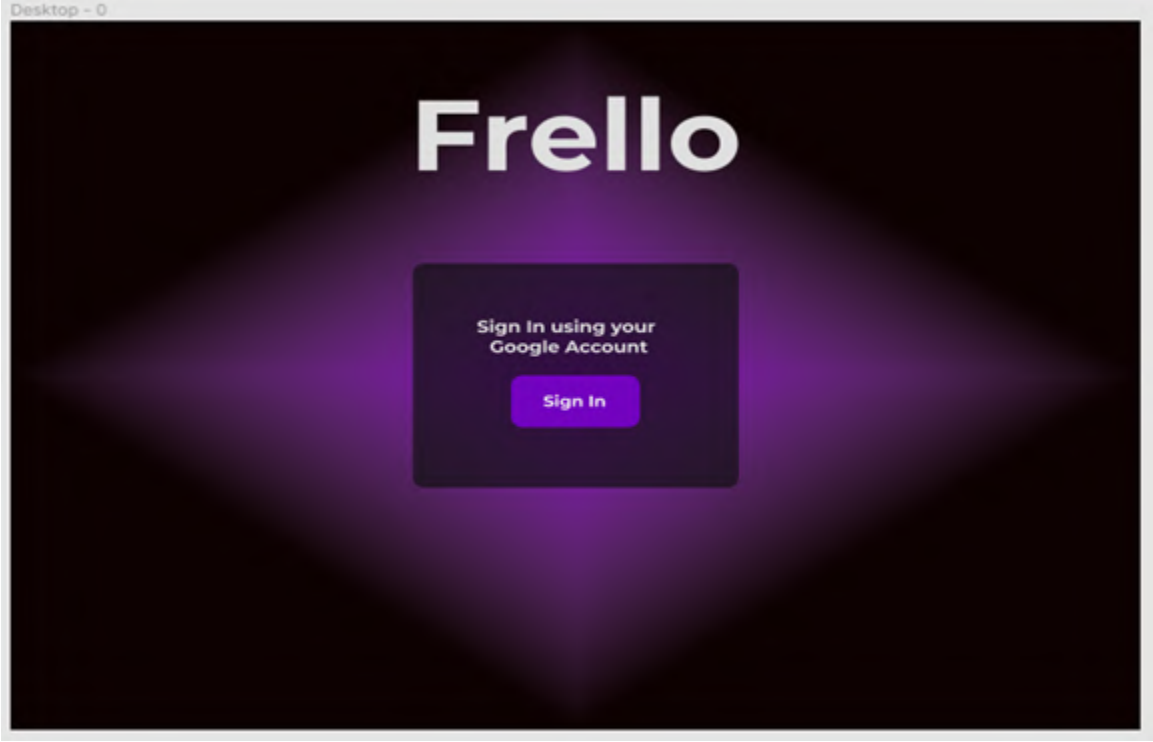


Figure 3, Sign in Page

Description

The Frello User Interface can be accessed via the link https://www.figma.com/file/Mog8az9aQL8EFFFB8mdE0L/S.E.-J-Component?node-id=0%3A1

The UI in figure 2 allows the user to Login with the help of his/her Google account. In the case of a new user who hasn't used our software before, Sign up option is also present. The user can use a google account which is not associated with any other individual's frello account to sign up.

4.1.3.2 Task and Subtask CRUD Module

User Interface Design

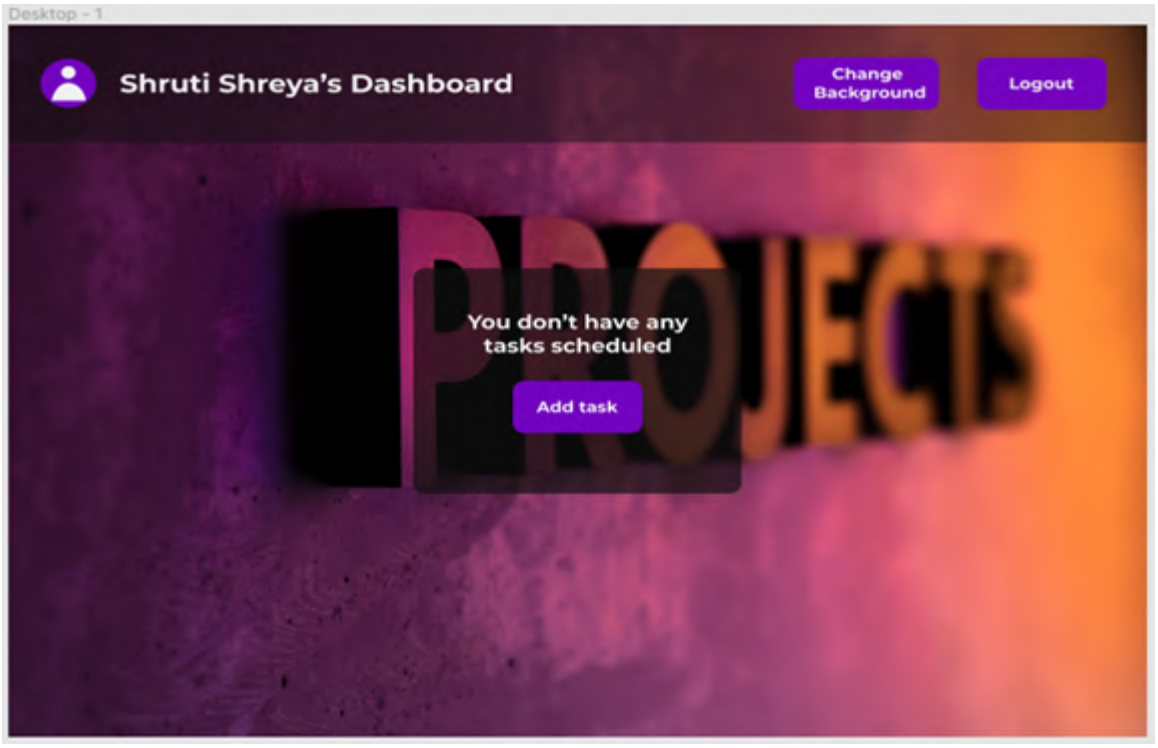


Figure 4, Default Dashboard

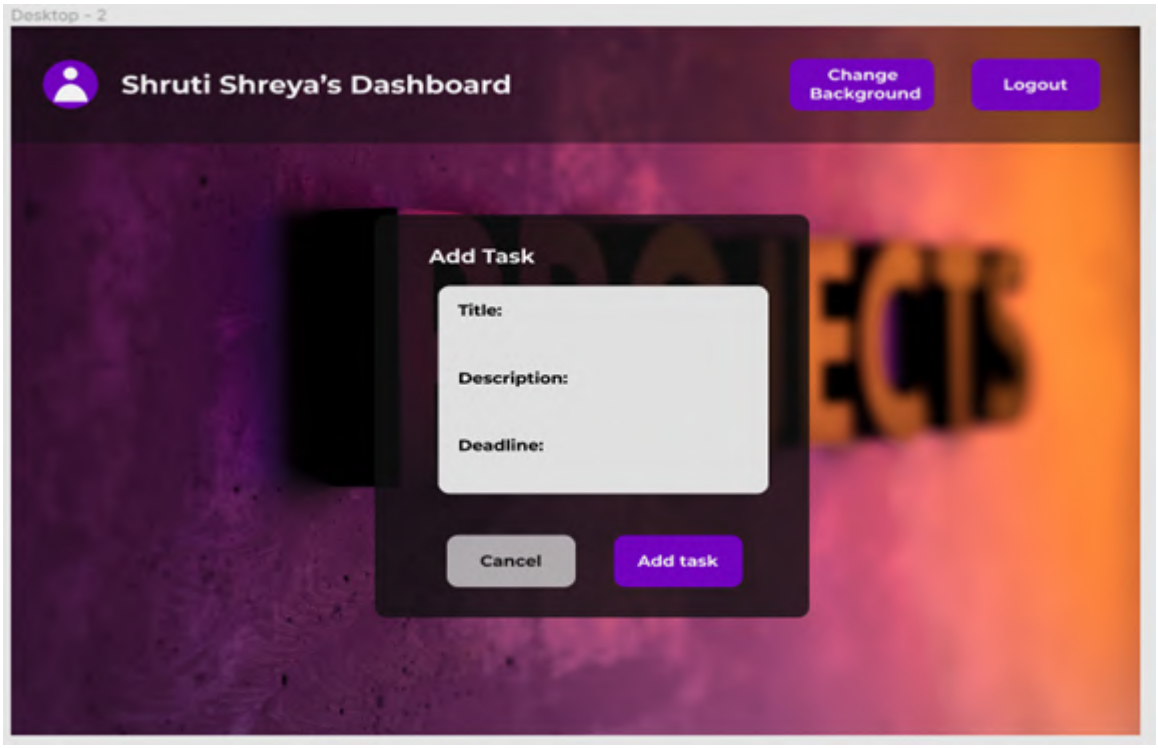


Figure 5, Add new Task

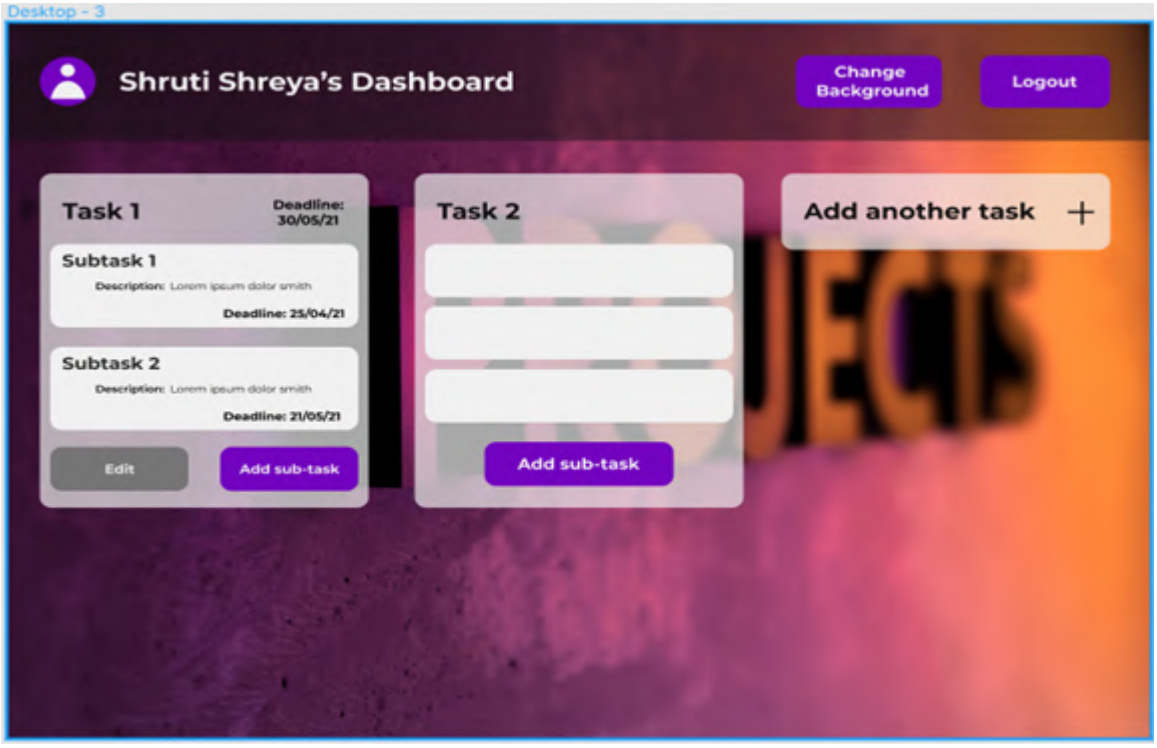


Figure 6, Customised dashboard after adding tasks and subtasks

Description

The User Interface allows the user to add, delete, replace, reorder and delete tasks according to their wish. The UI in figure 3 shows the default dashboard that the user will get after first successful login. After this step, the user can add tasks using the 'Add Task' button.

Once the button is clicked, the UI in figure 4 will open up asking the user to enter the task details. The user can enter that particular task's name, a short description as well as the deadline of the task. The'Add Task' button adds the task to the dashboard.

Figure 5 shows the user's customized dashboard with the tasks he/she has added. Subtasks can be added using the 'add subtask' button. Tasks/Subtasks can be edited/deleted. Re-ordering of tasks can be done by dragging.

4.1.3.3 Background Customisation Module

User Interface Design

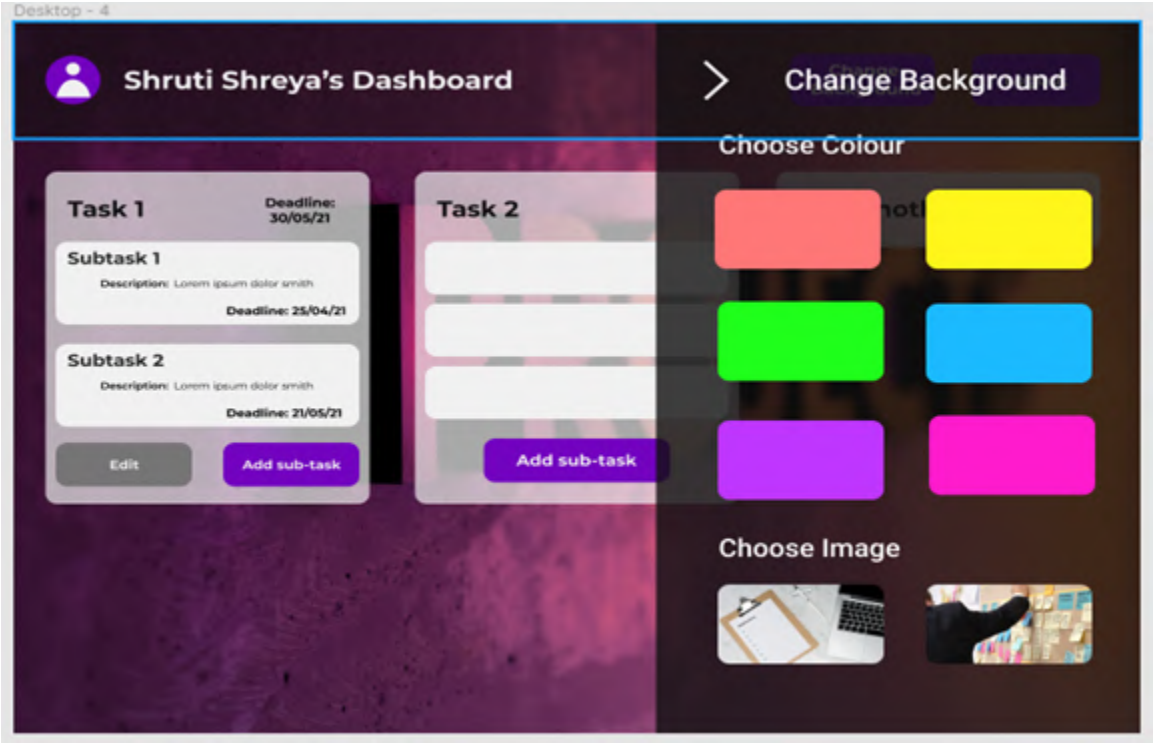


Figure 7, Customise background

Description

The user can change the background of their dashboard.

4.1.4 Architectural Design Model

We have used the Repository Model for our project.

The Sub-tasks for our project are:

1. Authentication Systems

2. Task CRUD

3. Sub-Task CRUD

4. Customization

4.1.4.1 Control Model

The timing of the event is not in control of the process in our project, which is why we are following an event based control model. Within which, we will be following the Broadcast model, so that when the user performs any operation such as creation or deletion of any particular task or sub-task, the subsystem involved and concerned with that particular task will only react and carry out the required actions.

4.1.2Use Case Diagram

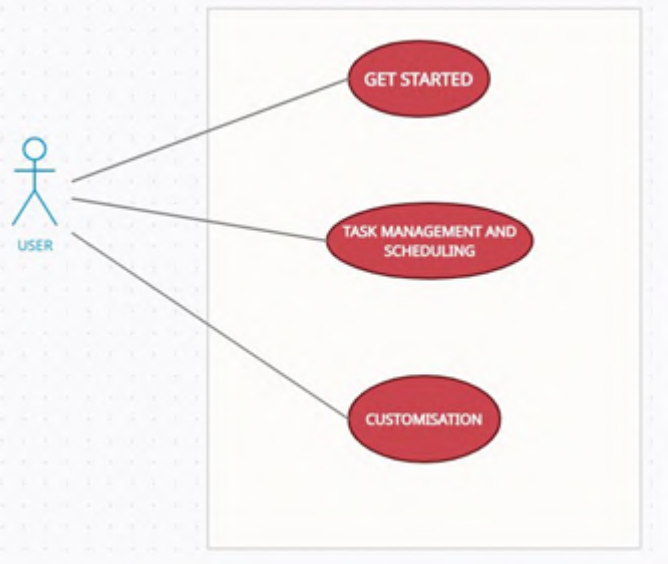


Figure 8, Use Case Diagram

4.1.5 Detailed Design

4.1.5.1 Module Detailed Design(Class Diagram)

Figure 9, Frello Class Diagram

4.1.5.2 Sequence Diagram

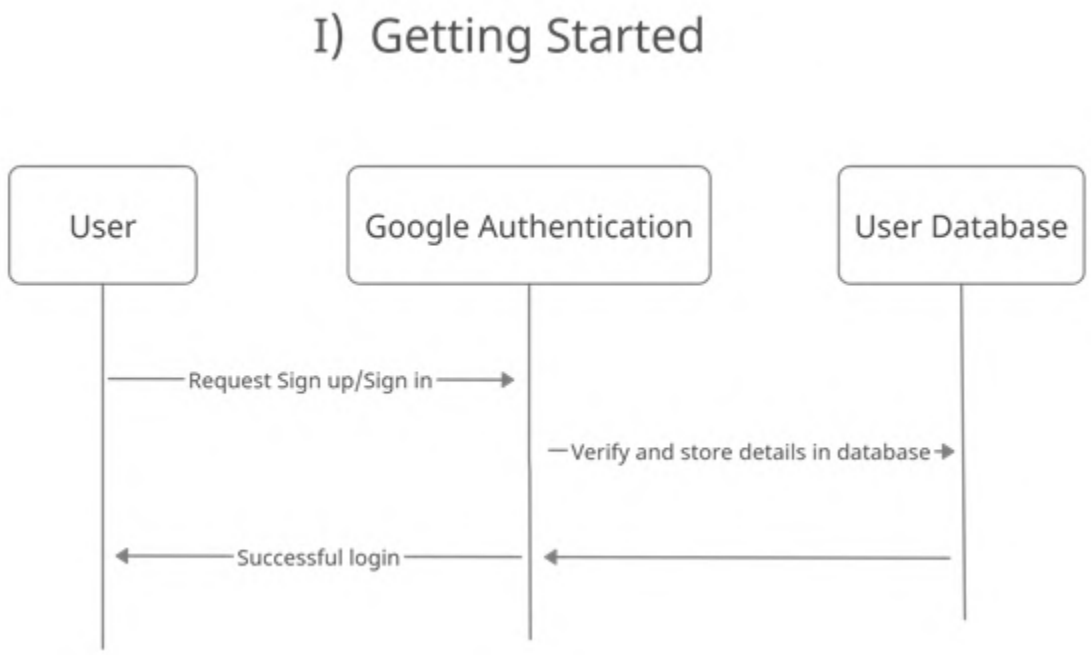


Figure 10, Sequence Diagram- Getting started

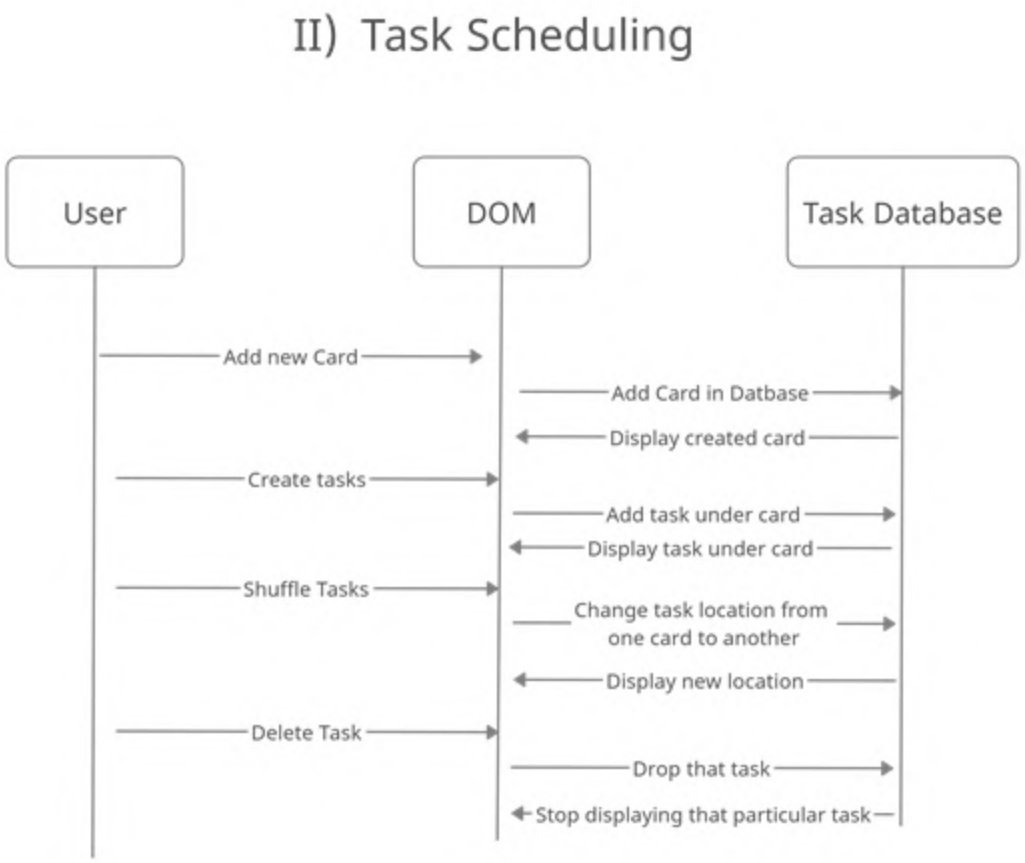


Figure 11, Sequence Diagram-Task Scheduling

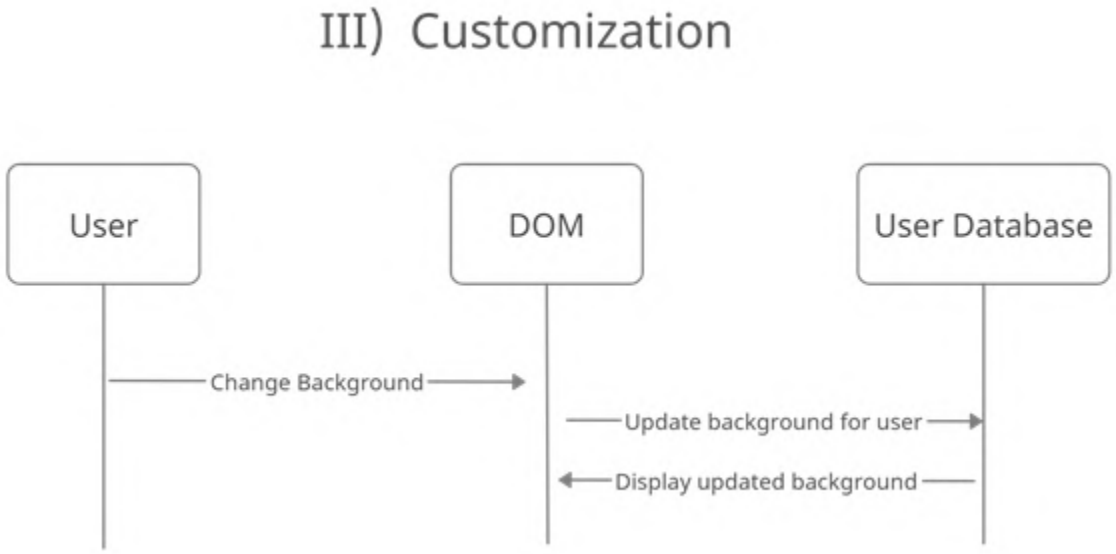


Figure 12, Sequence Diagram-Customisation

4.1.5.3 Collaboration Diagram

Figure 13, Collaboration Diagram- Getting started

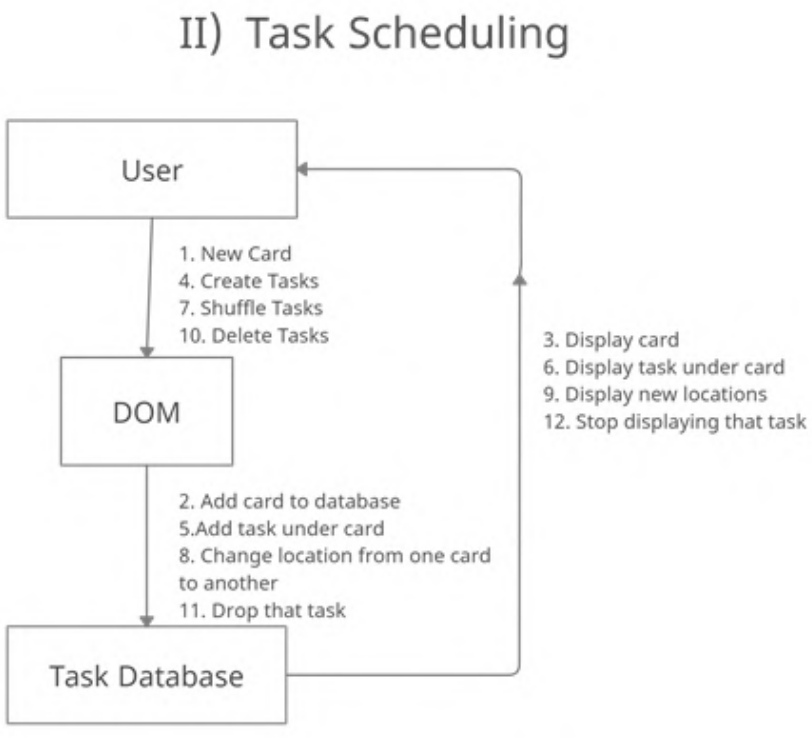


Figure 14, Collaboration Diagram- Task Scheduling

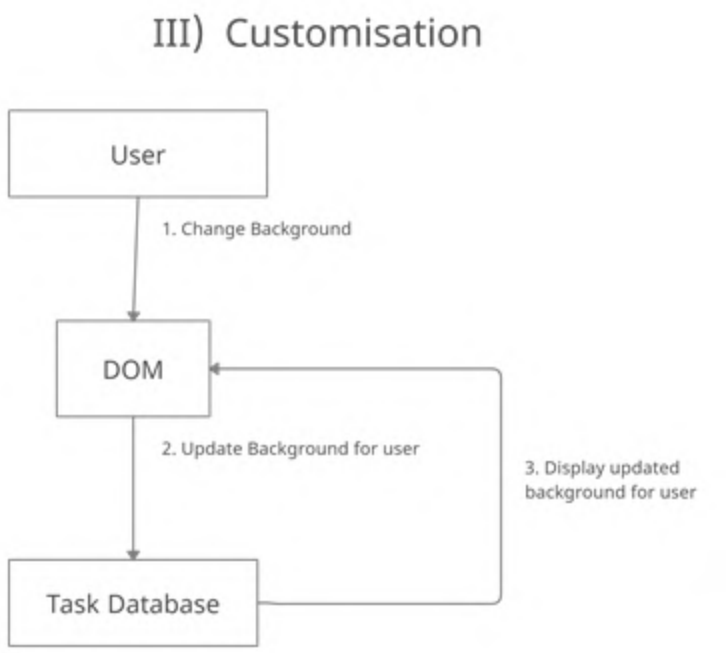


Figure 15, Collaboration Diagram- Customisation

4.1.5.3 Activity Diagram

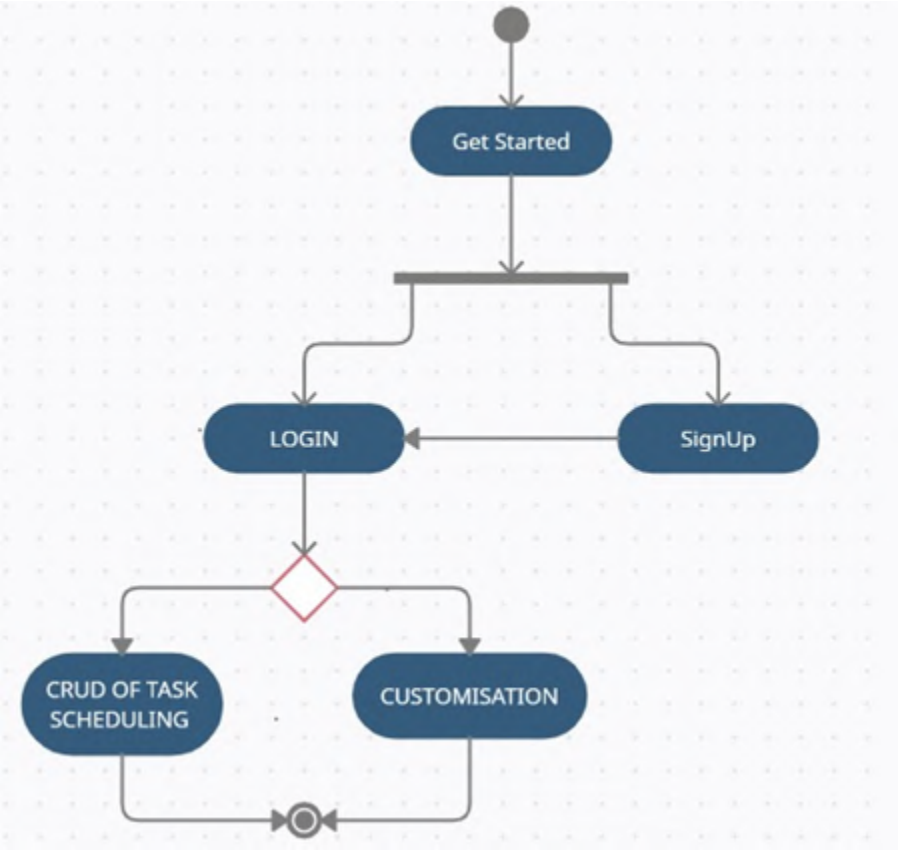


Figure 16, Activity Diagram

4.2 Codes and Standards

All project related codes have been uploaded on github.

https://github.com/Yash-Ray/Frello-SWE

4.3 Constraints, Alternatives and Tradeoffs

3.2 Constraints

The following is a table of the design constraints that the system shall meet.

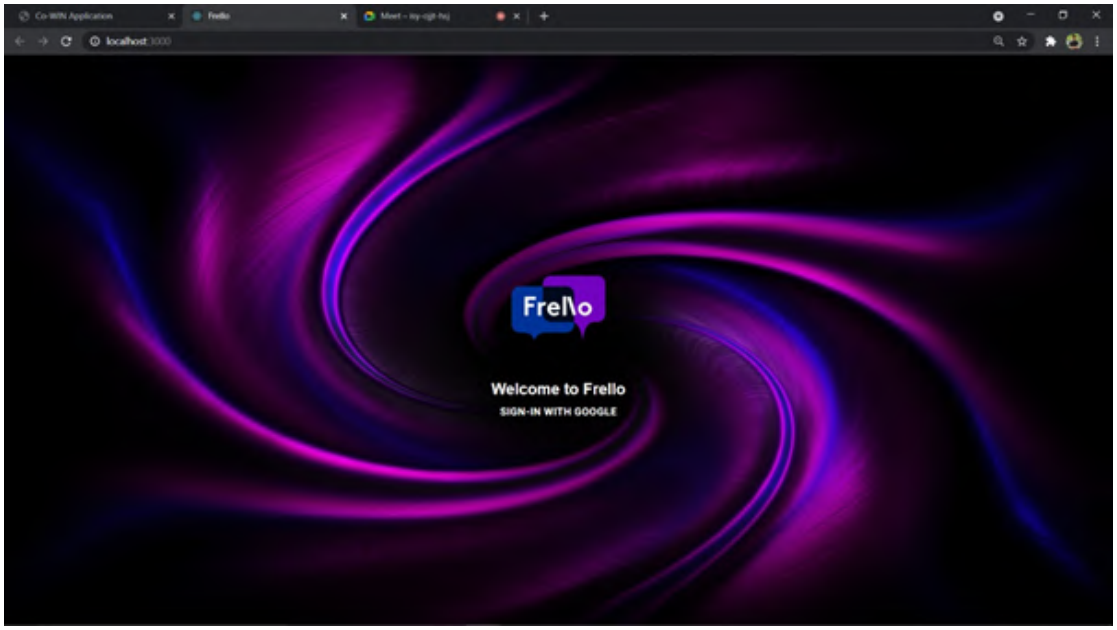
Table of Design Constraints

ID	Stakeholder	Requirement
1	User	The user shall only be able to set his/her personal goals and deadlines
2	Team members	The team members shall not view the user's individual progress or task distribution
3	User	The user shall provide his/her google account to facilitate authentication and login.
4	User	The user shall have a proper functioning browser

5. Software Testing

5.1 Authentication System Module

Test Case ID	Test Objective	Test Data	Expected Results	Actual Results	Test Pass/Fail
1.1.1	New User Sign Up	Add new Google Account details: Email: ray.yash.08@gmail.com Password: abc123	Redirect to User Dashboard	Redirect to User Dashboard	Pass
1.1.2	Existing user Sign In	Google Account details Email: ray.yash.08@gmail.com Password: abc123	Redirect to User Dashboard	Redirect to User Dashboard	Pass
1.2.0	Login details mismatch	Google Account details Email: ray.yash.08@gmail.com Password: abcabc	Display message: Wrong password	Display message: Wrong password	Pass

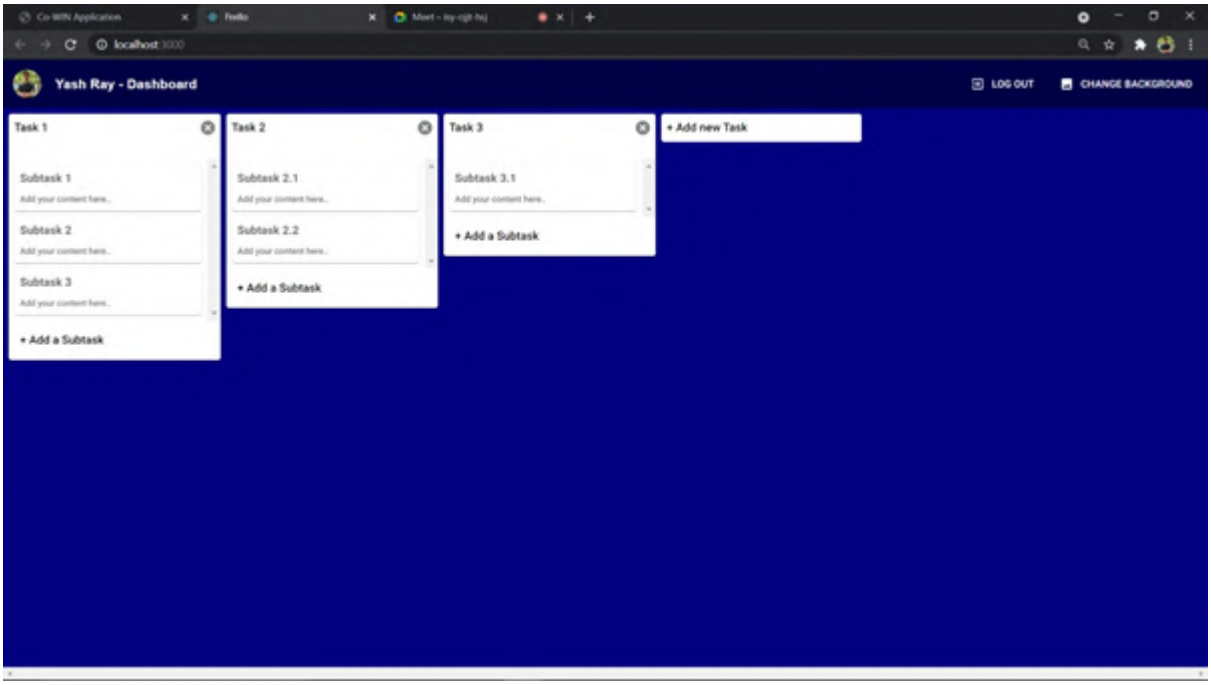


Authentication via Google account (Home Page)

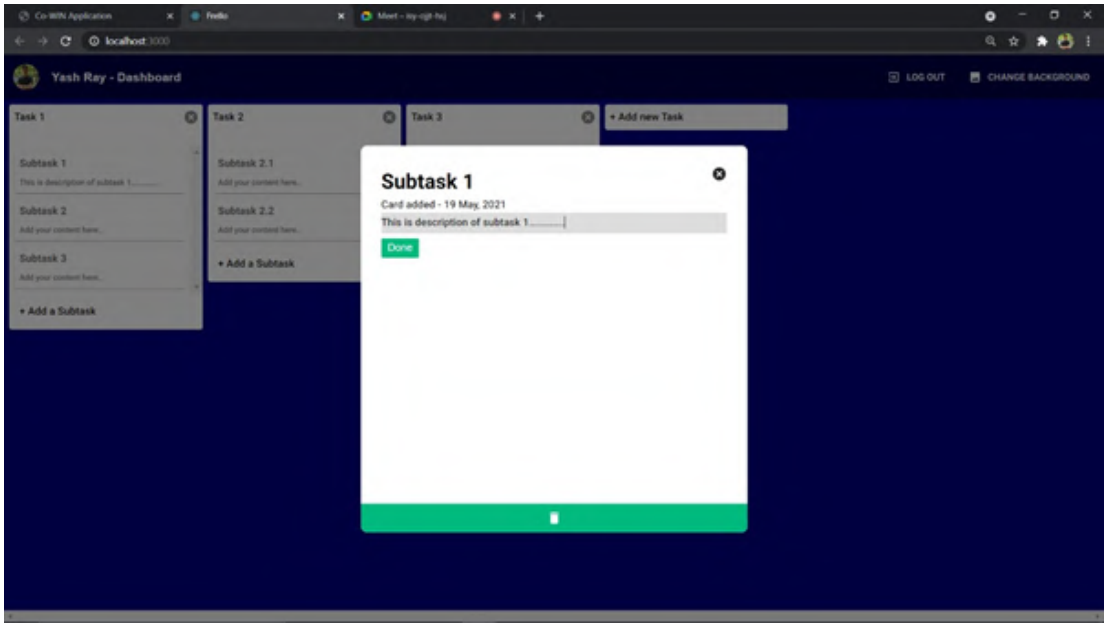
5.2 Task and Subtask Module

Test Case ID	Test Objective	Test Data	Expected Results	Actual Results	Test Pass/ Fail
2.1.0	Add new task	Clicking the “Add new task” button	Displaying “Add title” input box	Displaying “Add title” input box	Pass
2.2.0	Add task title	Title: Task 1	Displaying a new list with title “Task 1” on the Dashboard	Displaying a new list with title “Task 1”	Pass
2.2.1	Delete a task	Clicking the “x” button next to Task 1	Task 1 disappears from Dashboard	Task 1 disappears from Dashboard	Pass
2.3.0	Add subtask	Clicking the “Add new Subtask” button below Task 1	Displaying “Add title” input box	Displaying “Add title” input box	Pass

2.3.1	Add subtask title	Title: Subtask 1	Display new subtask under Task 1 with title ‘Subtask 1’	Display new subtask under Task 1 with title ‘Subtask 1’	Pass
2.3.2	Open subtask options	Clicking on ‘Subtask 1’	Display pop up with subtask options	Display pop up with subtask options	Pass
2.3.3	Add/edit subtask details	Subtask details: ‘This task is supposed to… ‘	Display subtask details under ‘Subtask 1’	Display subtask details under ‘Subtask 1’	Pass
2.3.4	Delete subtask	Click the ‘delete’ button in subtask options	‘Subtask 1’ disappears from Task 1	‘Subtask 1’ disappears from Task 1	Pass
2.3.4	Close subtask options	Click the ‘x’ button in subtask options	Subtask options closes and Dashboard appears	Subtask options closes and Dashboard appears	Pass



User Dashboard (Task/Subtask creation, organisation and deletion)



Subtask Options (add description, delete subtask)

6. Schedule, Tasks and Milestones

6.1 Work Breakdown Structure

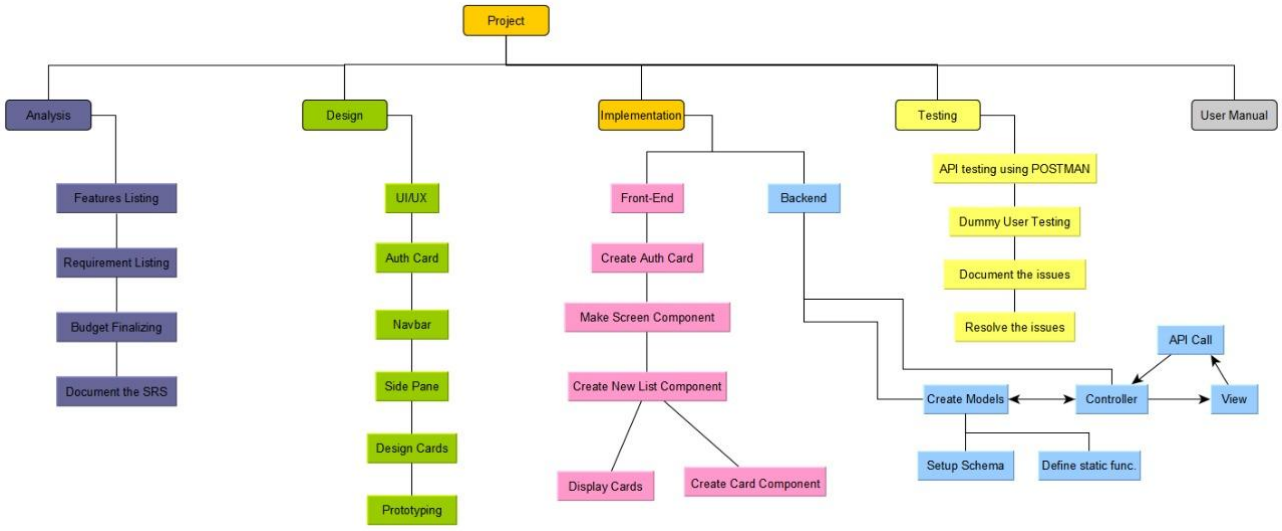


Figure 17, Work Breakdown Structure

6.2 Gantt Chart

6.2.1 Process Based Gantt chart

i) Task Names

Sr. no.:	Tasks
1	Analysis
1.1	Features Listing
1.2	Requirement Listing
1.3	Budget Finalizing
1.4	Document the SRS
2	Design
2.1	Page 1
2.2	Page 2
3	Implementation
3.1	Front-end
3.1.1	Page 1
3.1.2	Page 2
3.2	Back-end
3.2.1	Models
3.2.2	API
3.2.3	View
3.3	Integration
4	Testing
4.1	API Testing
4.2	User Testing
4.3	Documenting issues
4.4	Resolution of issues
5	User Manual

Figure 18, Task details of process based gantt chart

ii) Gantt chart

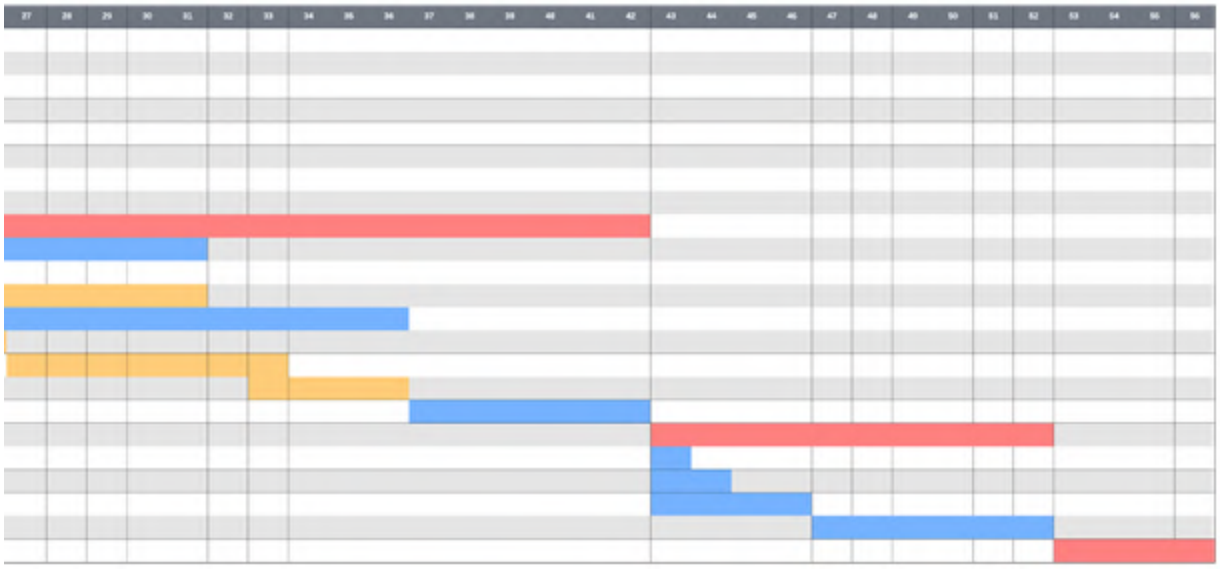


Figure 19, Process based gantt chart

6.2.2 Product Based Gantt Chart

i)Task Names

Sr. no.:	Tasks
1	Design
1.1	Page 1
1.2	Page 2
2	Implementation
2.1	Front-end
2.1.1	Page 1
2.1.2	Page 2
2.2	Back-end
2.2.1	Models
2.2.2	API
2.2.3	View
2.3	Integration
3	Testing
3.1	API Testing
3.2	User Testing
3.3	Documenting issues
3.4	Resolution of issues

Figure 20, Task details of product based gantt chart

ii)Gantt Chart

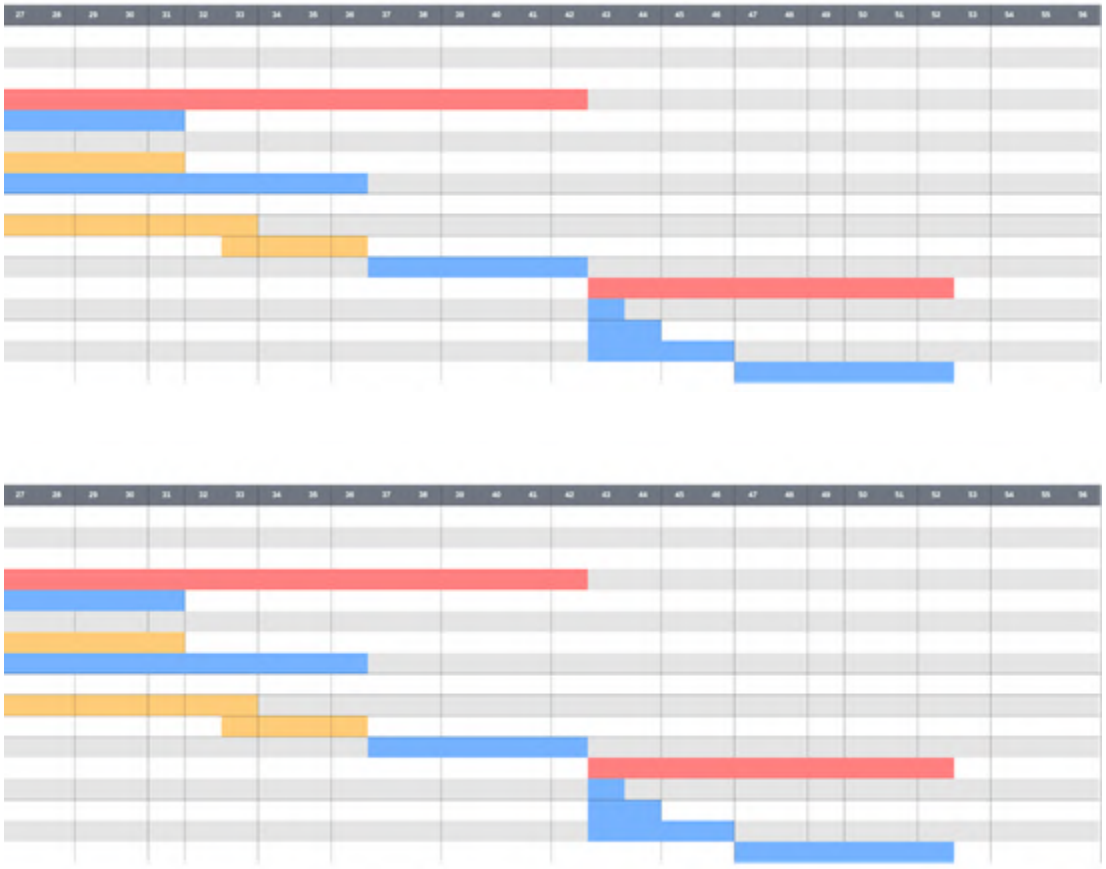
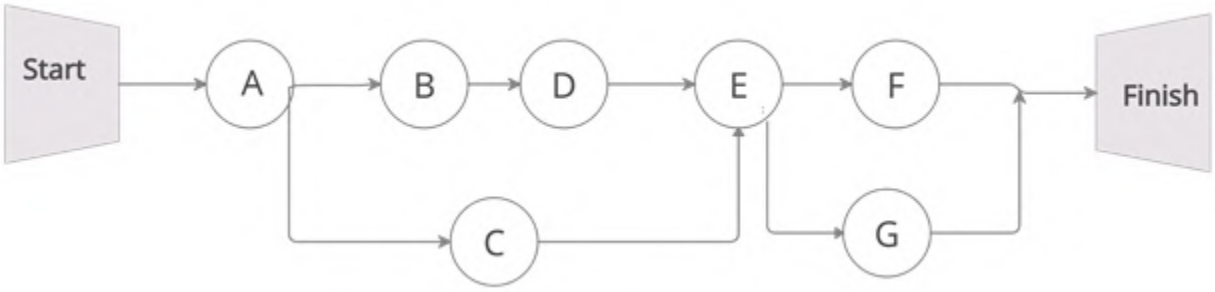


Figure 21, Product based gantt chart

6.3 Activity Network

Task	Label	Predecessors	Staff Required	Estd. Duration (days)
Analysis	A	-	3	7
Design	B	A	2	7
Backend	C	A	1	28
Frontend	D	A,B	2	21
Integration	E	A,B,C,D	3	14
Testing	F	A,B,C,D,E	2	7
Preparing User Manual	G	A,B,C,D,E	1	4



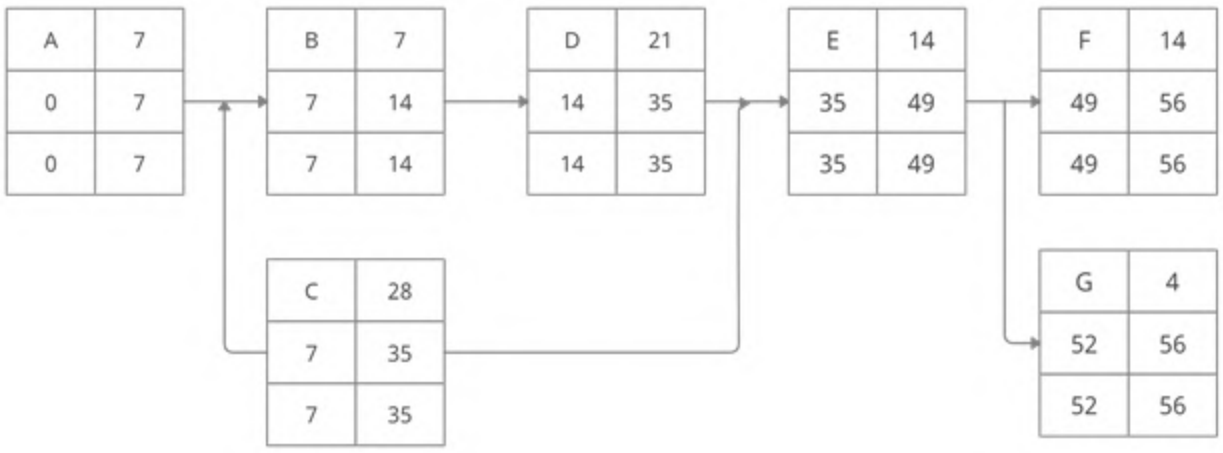


Figure 22-24, Activity Network

6.4 Activity Timeline

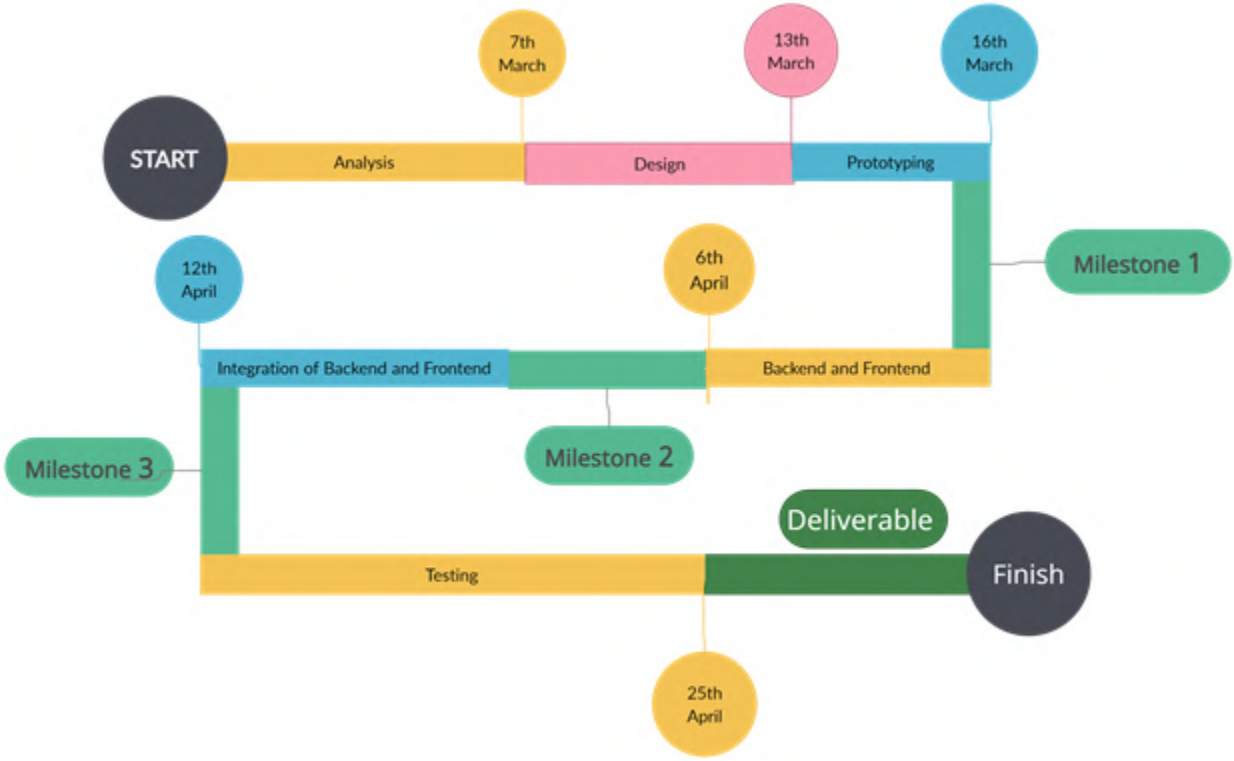


Figure 25, Activity Timeline

7. Project Demonstration

7.1 Authentication Module

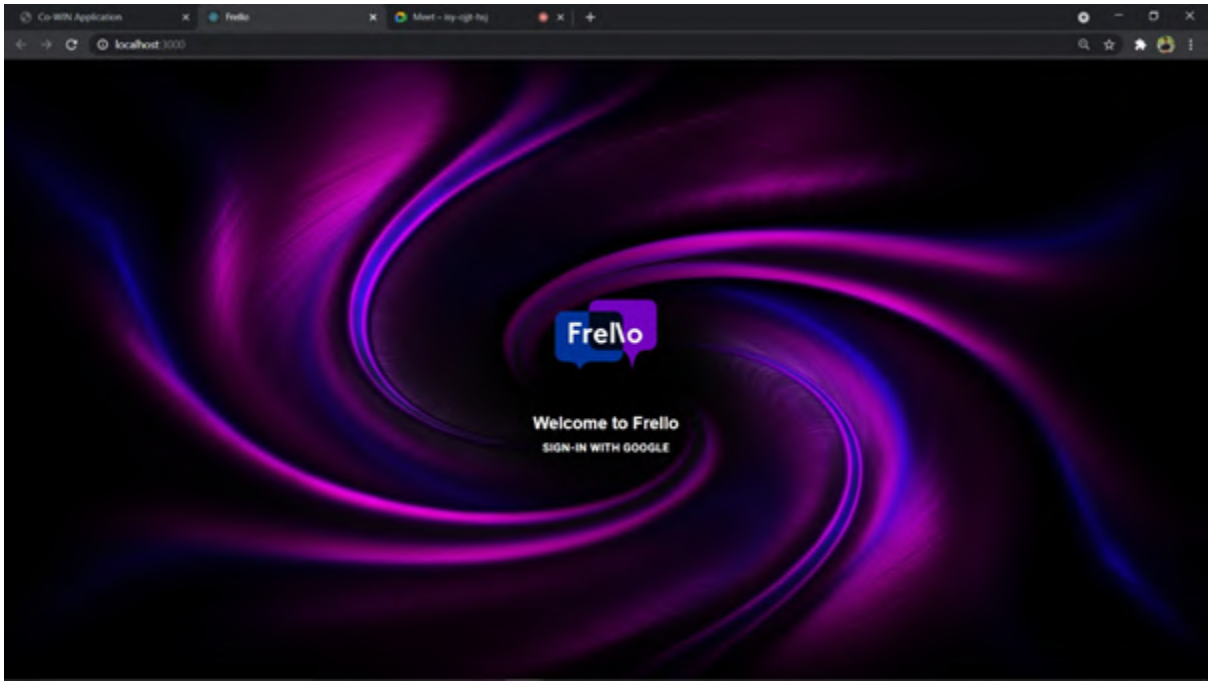


Figure 26: Sign in/Sign up via Google Account (Home Page)

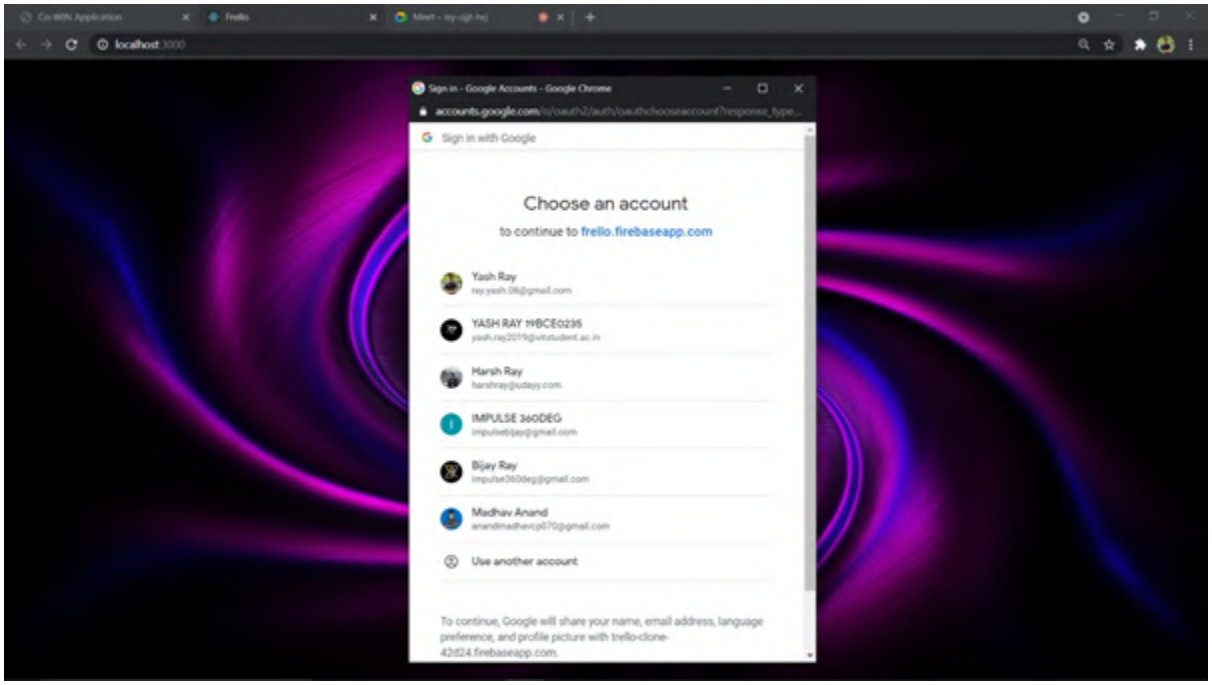


Figure 27: Google account Authentication

7.2 Task/Subtask Module

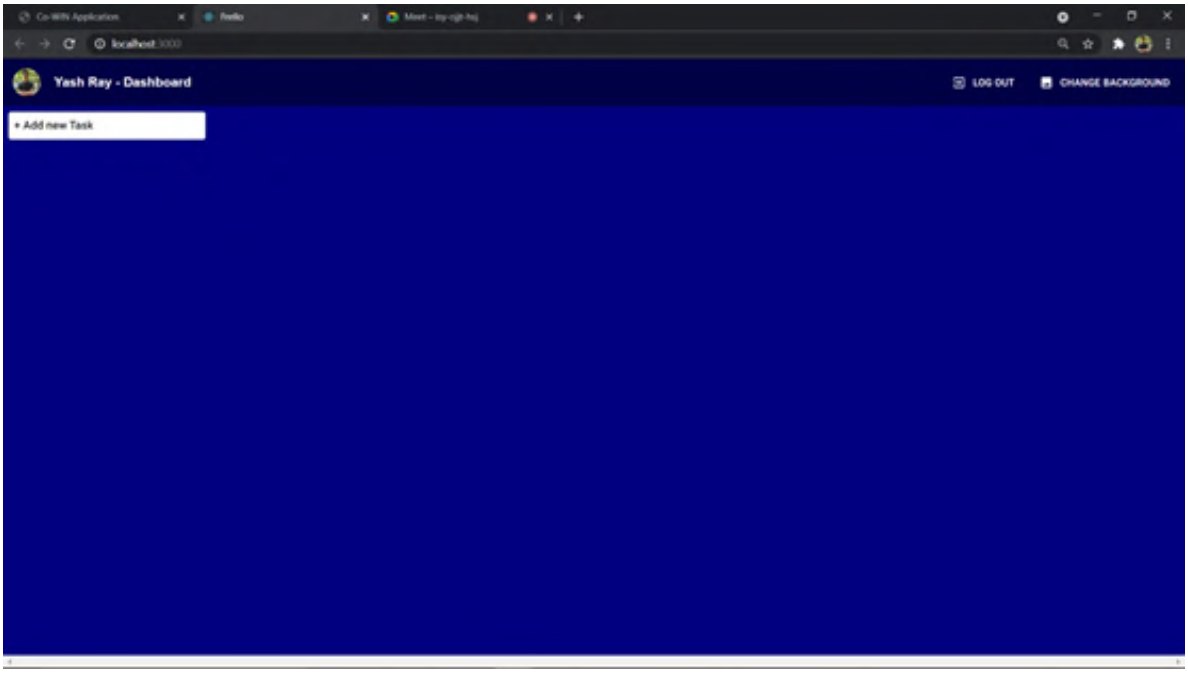


Figure 28: User Dashboard after first login.

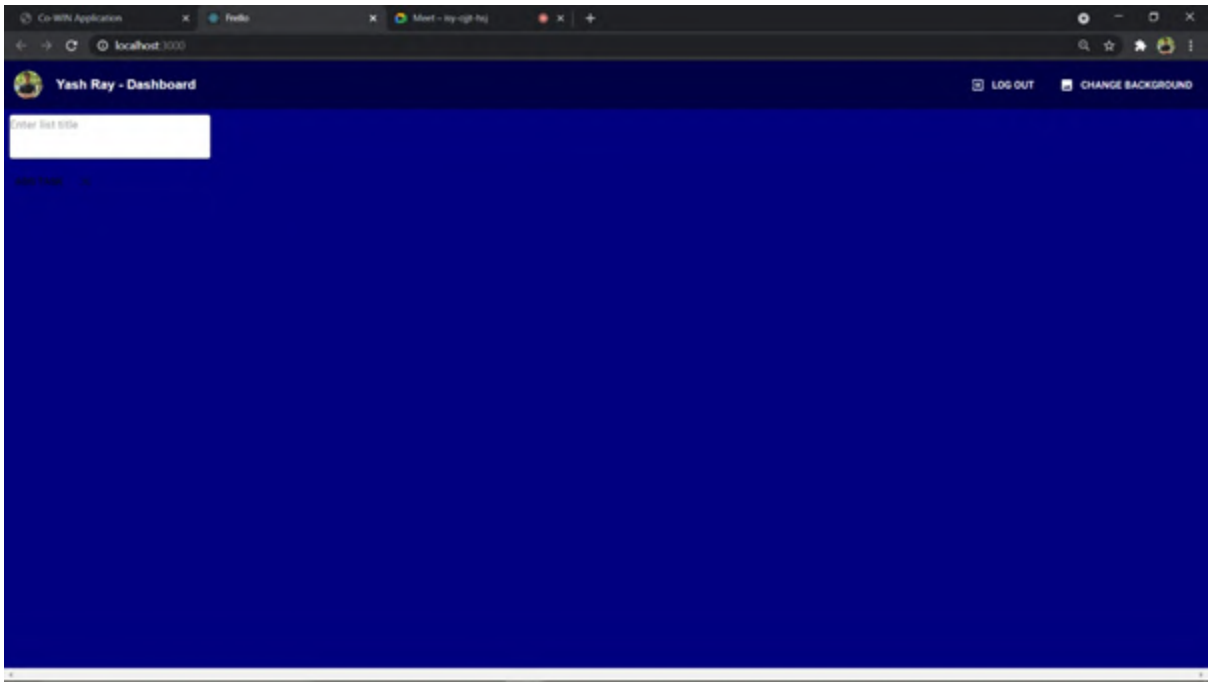


Figure 29: Status after clicking ‘Add new Task’ button. (Input box opens to enter task title)

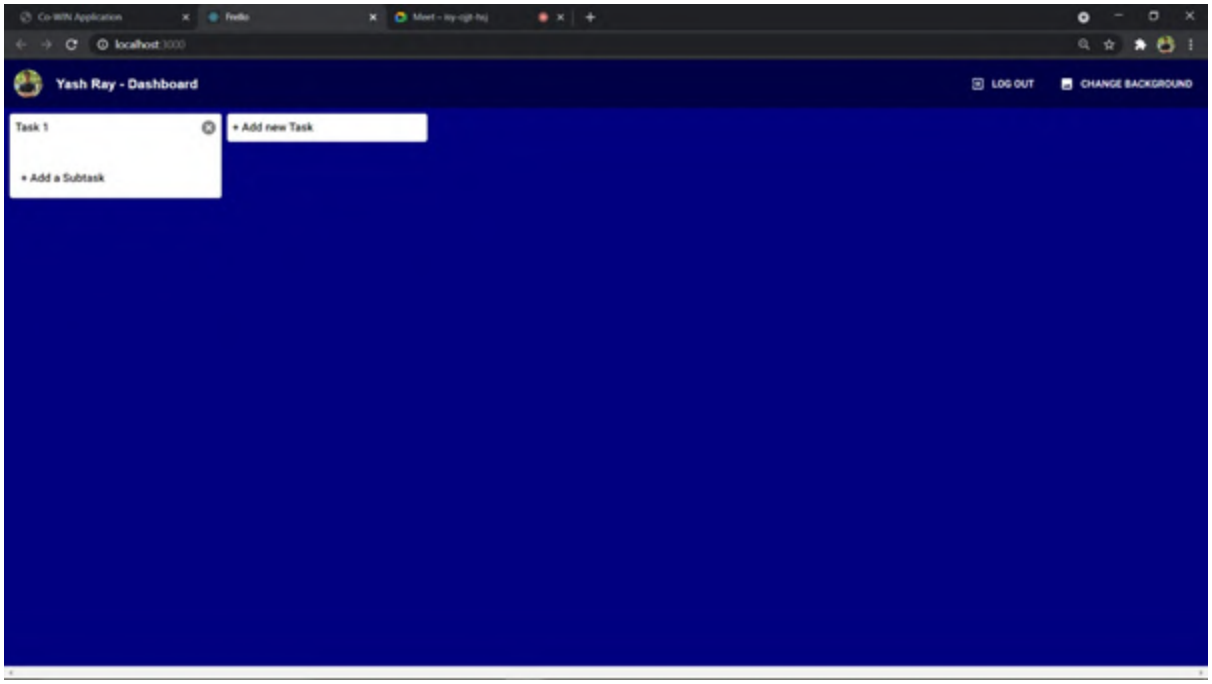


Figure 30: Status after first task creation (More tasks can be added independently and subtasks can be added under those tasks)

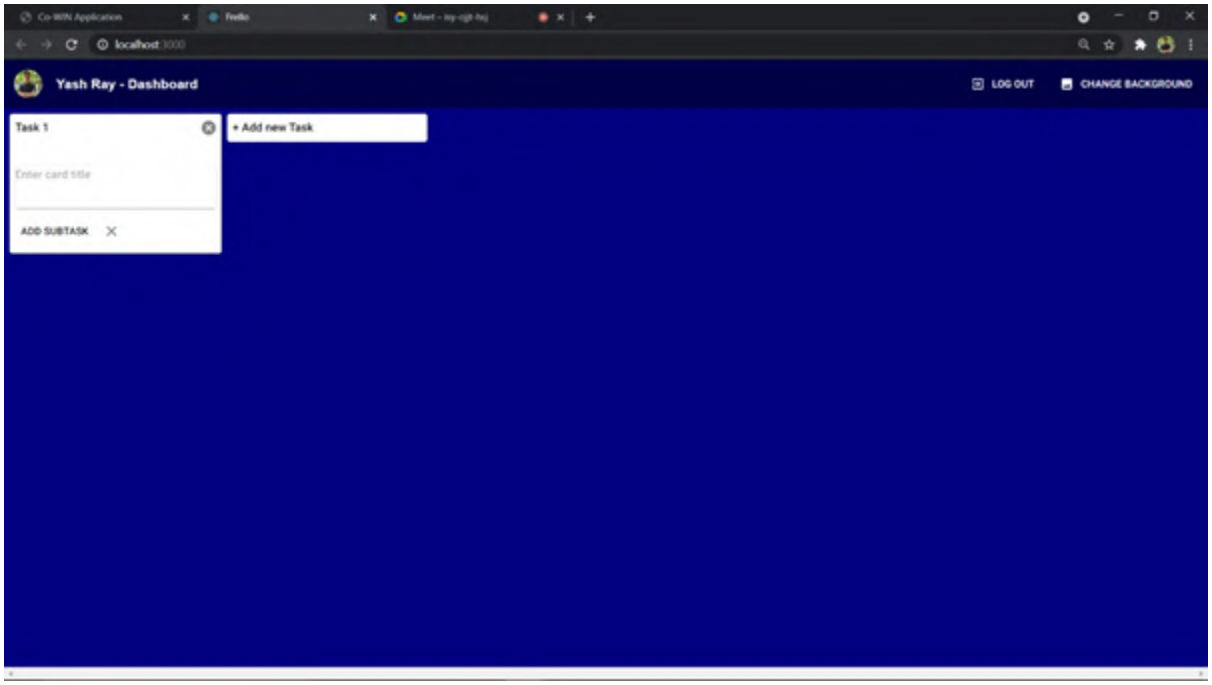


Figure 31: Status after clicking 'Add a Subtask' button under Task 1 (Input box opens to add subtask title)

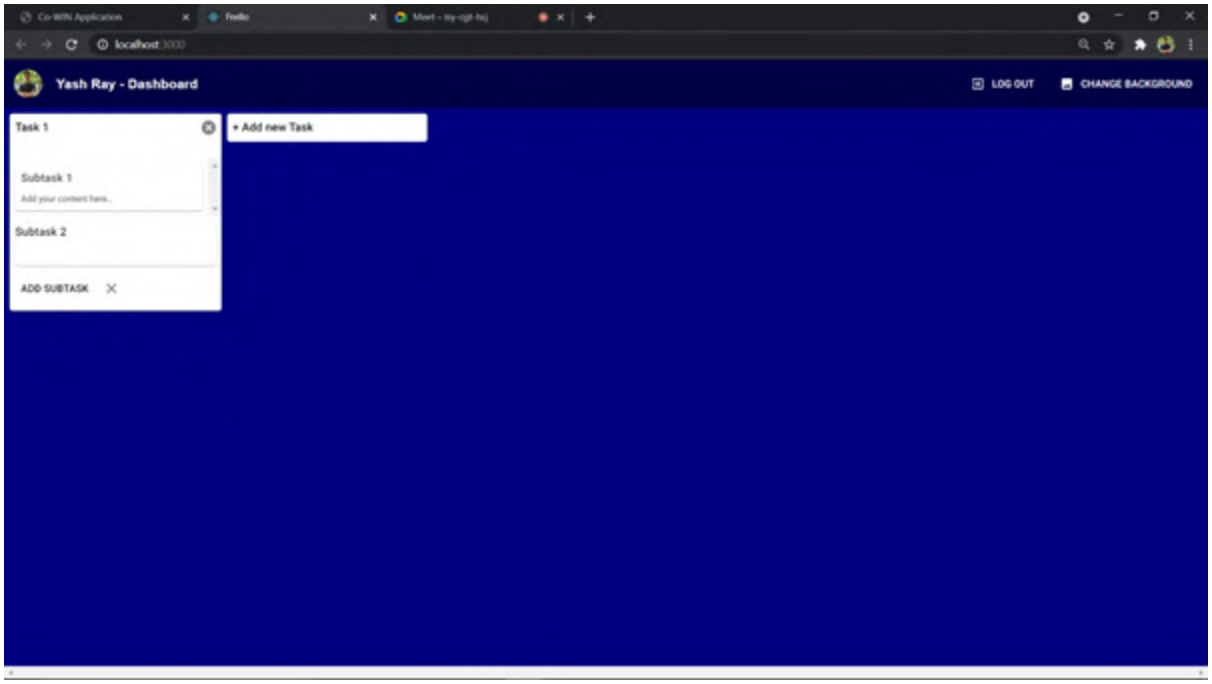


Figure 32: Adding more subtasks under Task 1

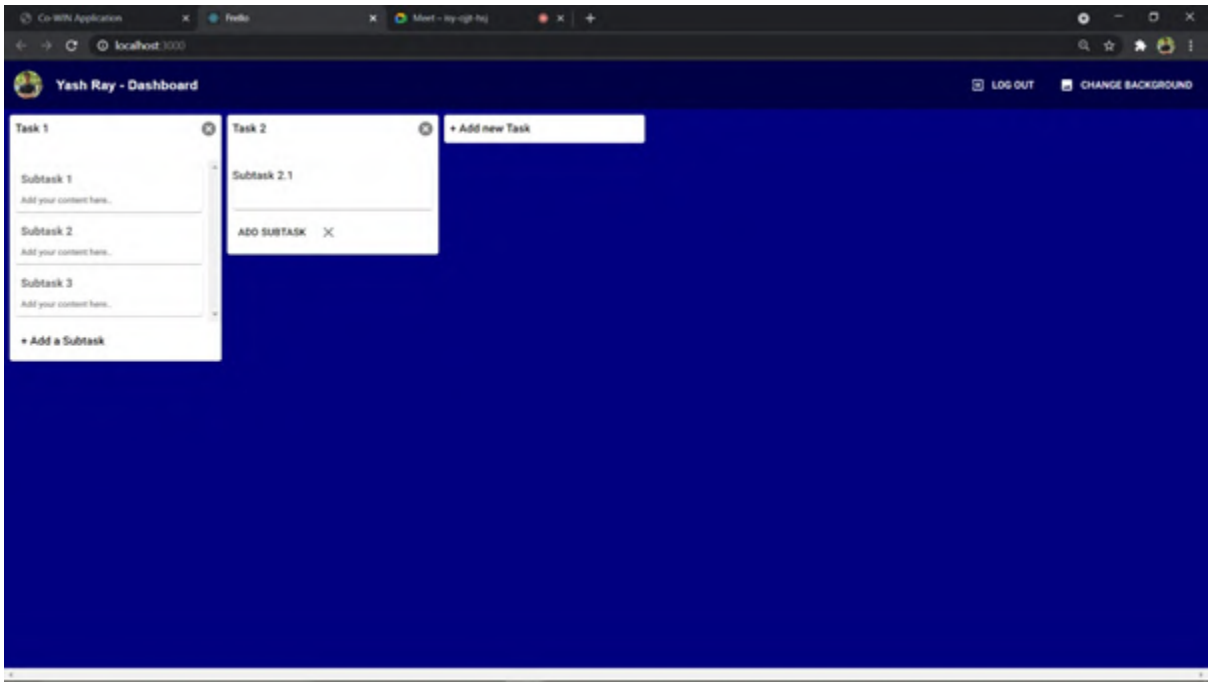
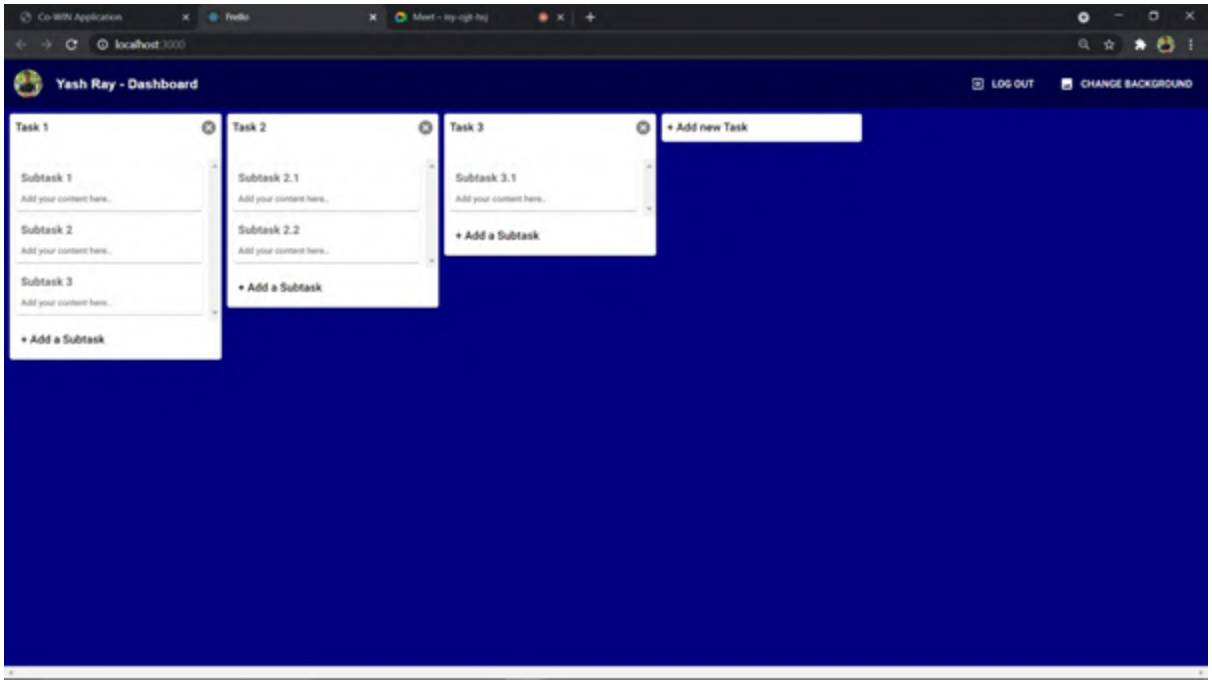


Figure 33: Adding multiple Tasks



Firgure 34: Adding multiple subtasks under different tasks

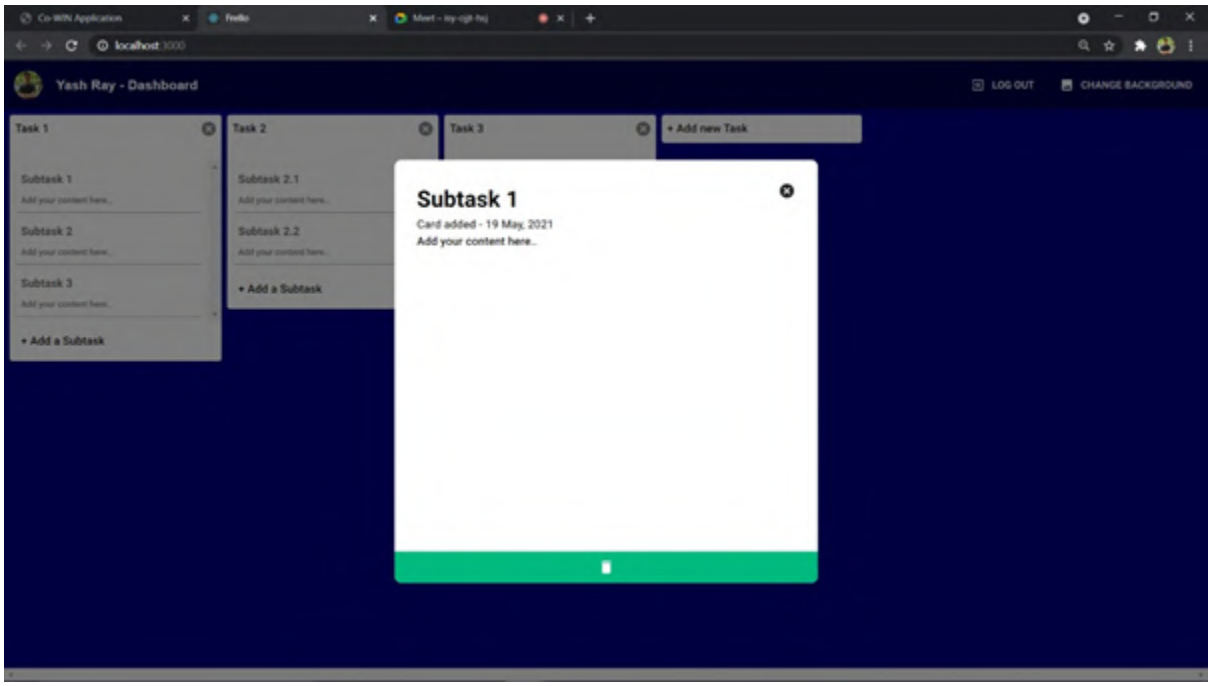


Figure 35: Status after clicking on any subtask(Subtask 1 in this case)

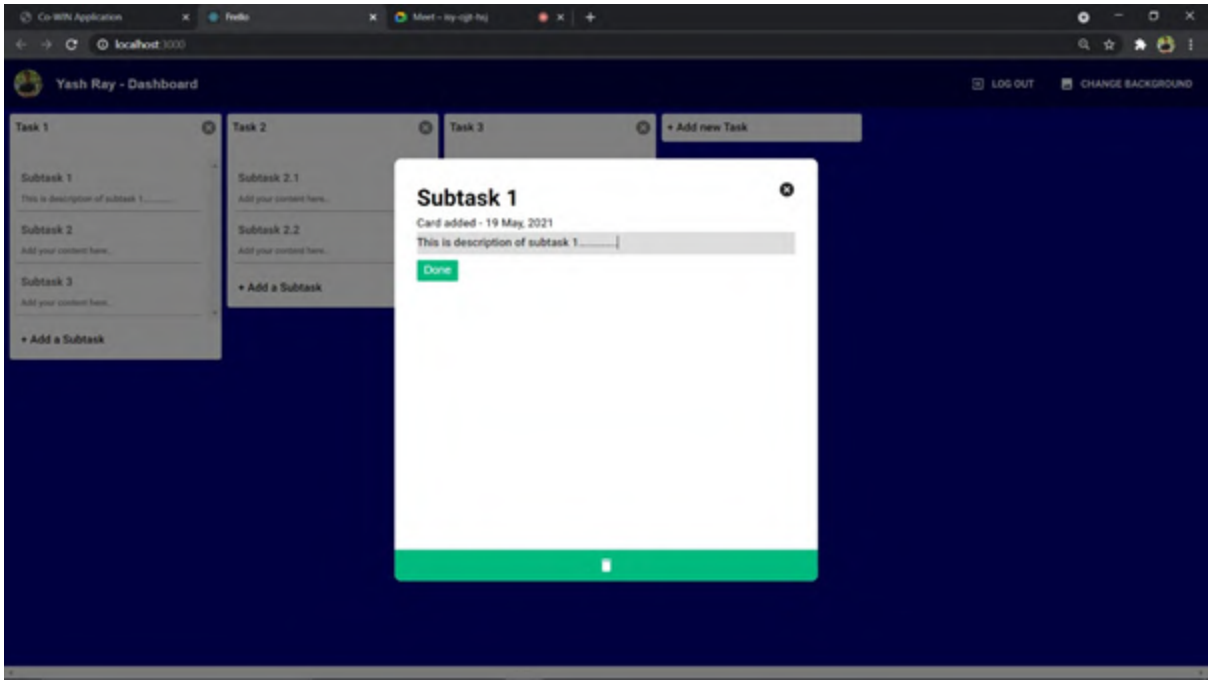


Figure 36: Adding description to Subtask 1.

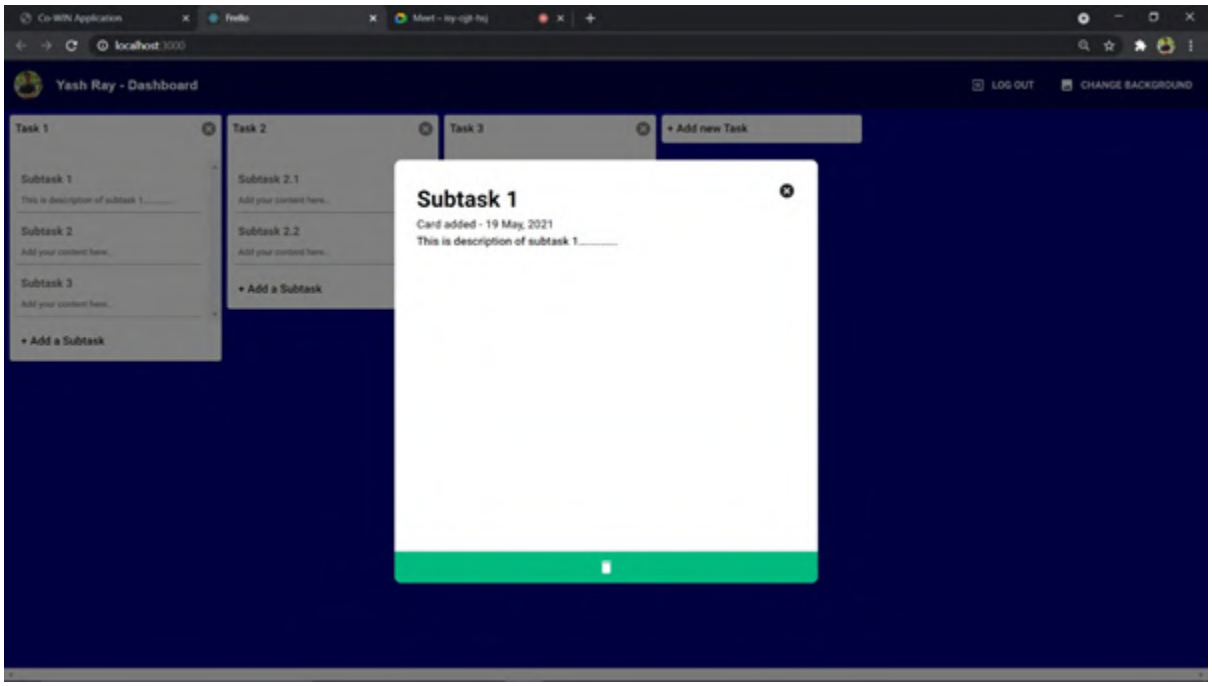


Figure 37: Description added. Subtask can be deleted using the delete button and subtask options can be closed using ‘x’

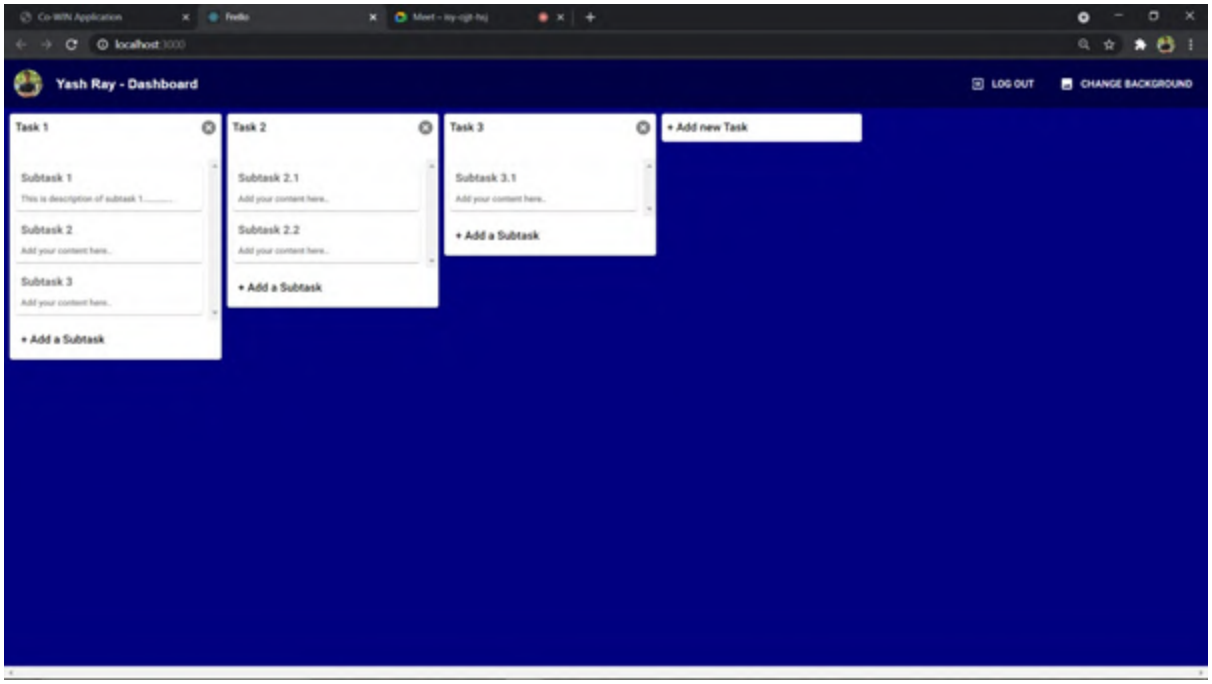


Figure 38: Complete Dashboard with Tasks, Subtasks and their description

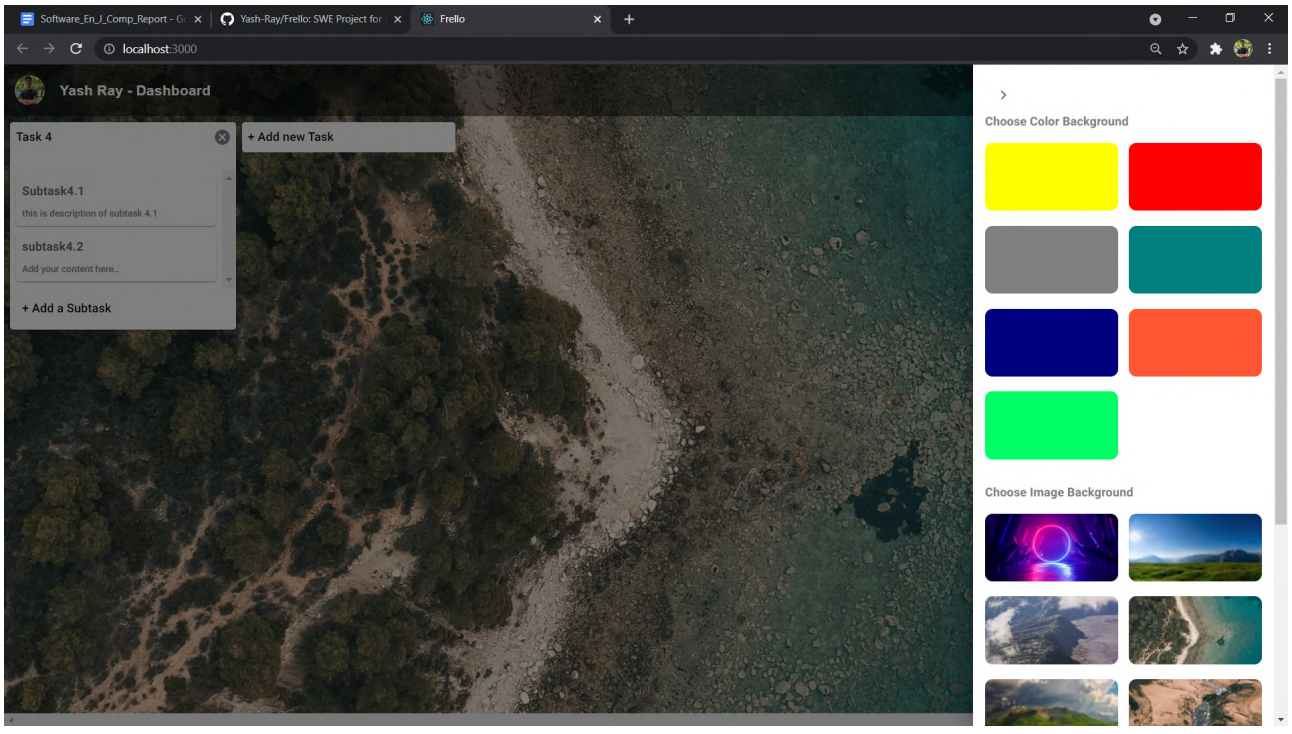


Fig 39: Side-Pane view having options for background

8. Concluding Remarks

Frello- project management tool was developed in approximately 35-40 hours. This included time taken for research, planning, documentation and development. The project documentation and analysis alone took 15-20 hours. It's development involved contributions from all the team members.

The project can be taken in the future again for adding more advanced features like Deadline/Due date and time and changing of task stickers color based on priorities. As of now, all the software development conditions have been met and the project can be deployed in public. Users can sign up and use Frello for task scheduling and organizing.

Suggestions or requests are accepted and will be adhered after ample discussions and possibilities.

.