Java Full Stack Phase1 Project

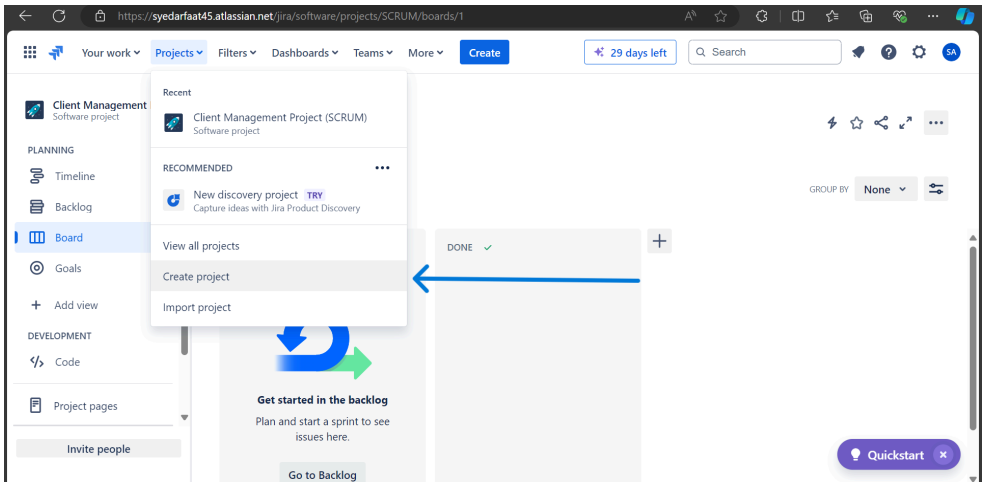
Name: Syed Arfaat

Project Name: Client Management System

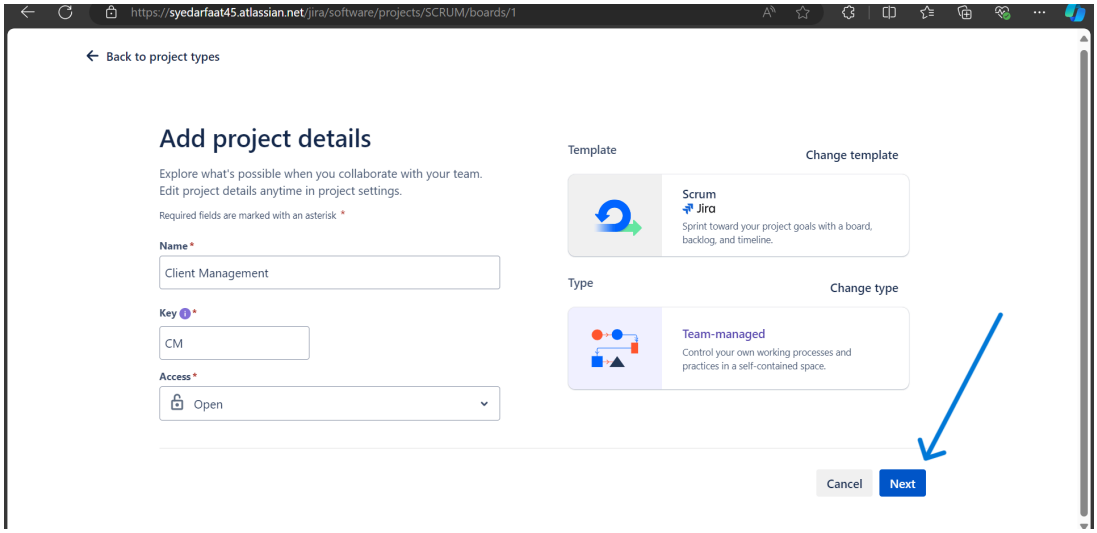
Ouput Screenshots:

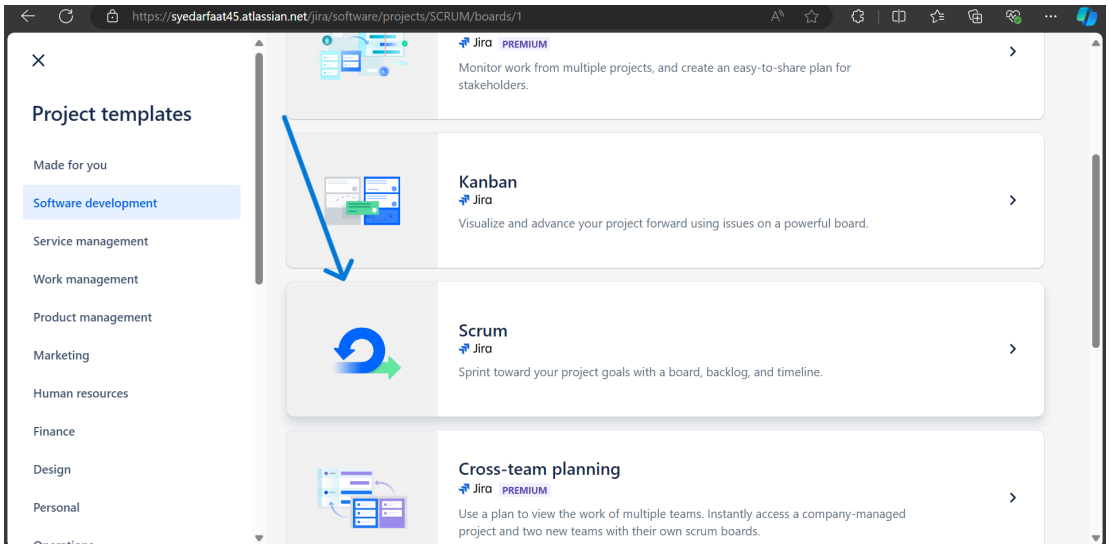
Agile Related Screenshots:

Creating a new project on jira after setting up an account.

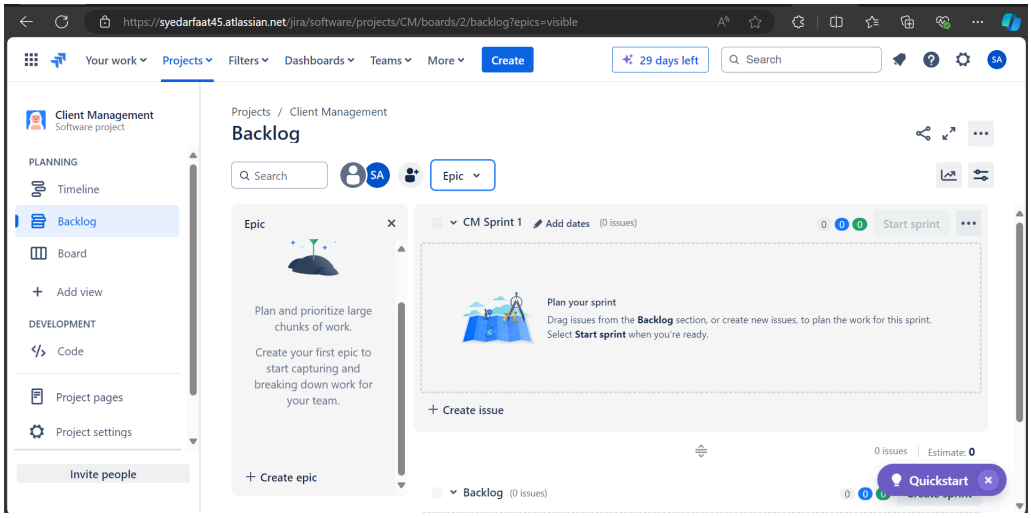


Creating a new scrum project:

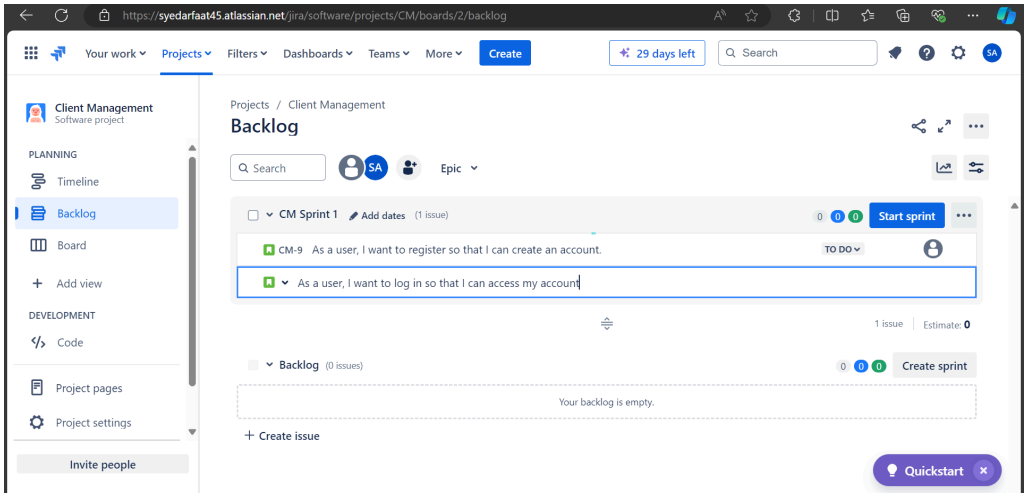




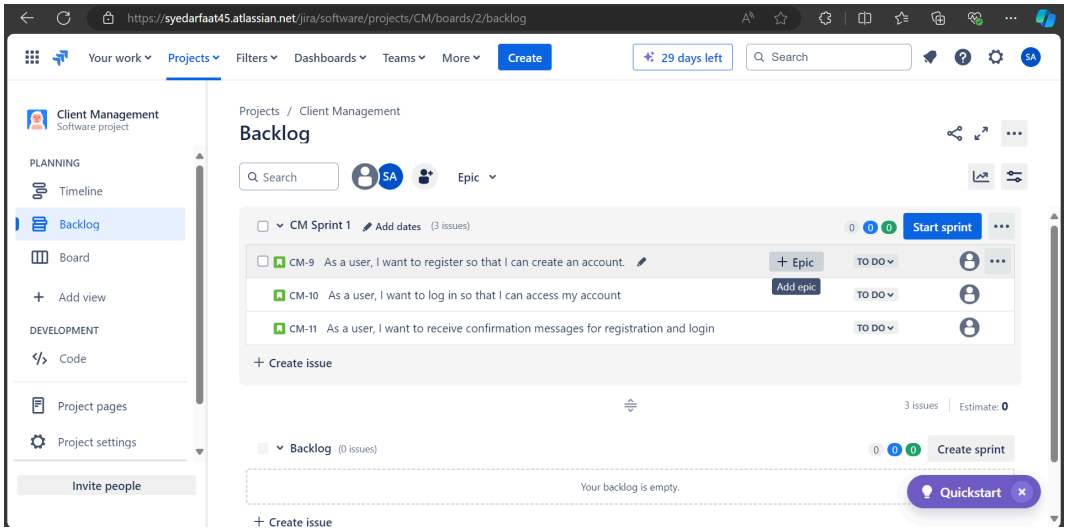
This is how the dashboard looks like after creating a scrum.

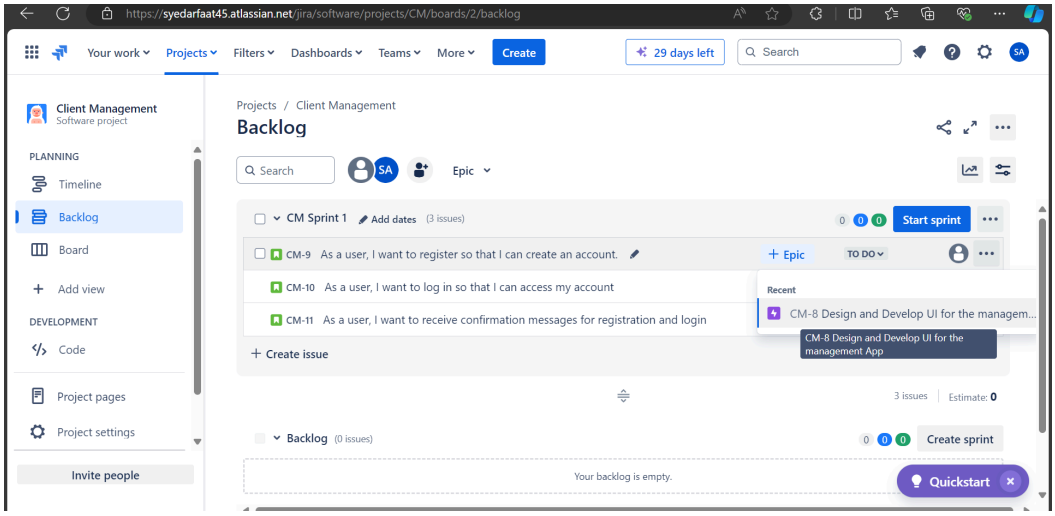


The user now creates user stories based on their requirements.

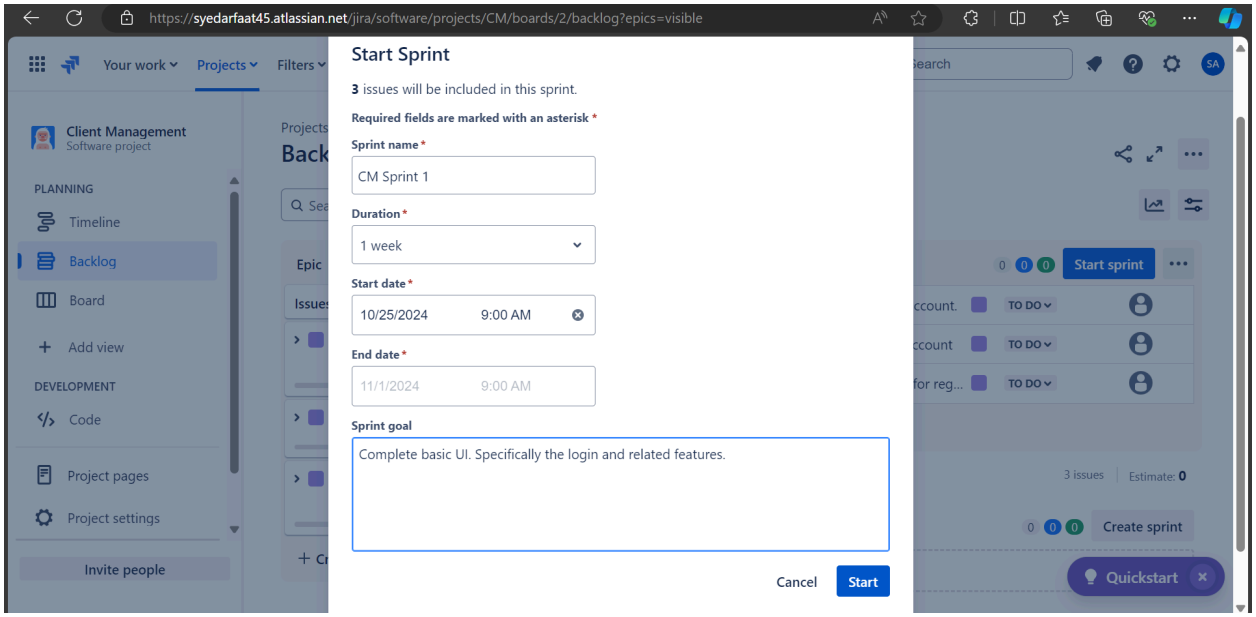


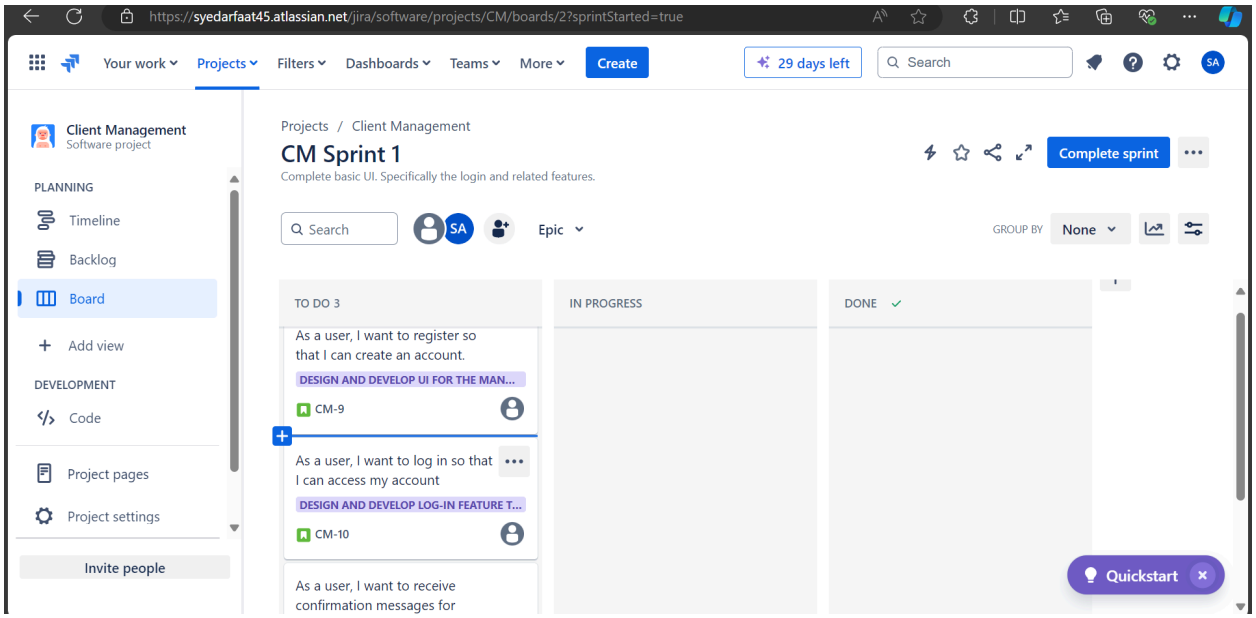
Based on user requirements, an epic is created to meet these requirements by the team and these epcis are then assigned to to members of the team.



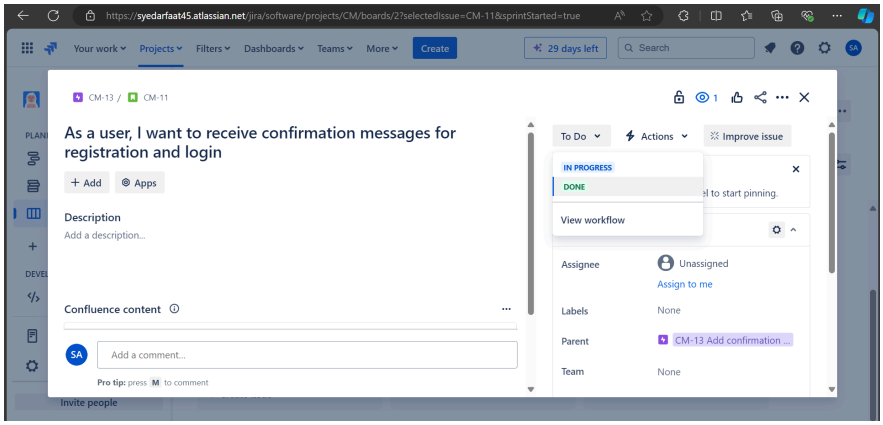


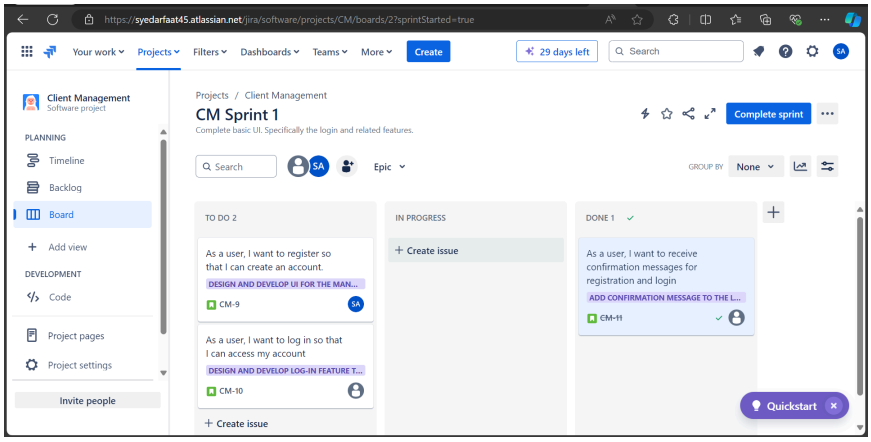
And then sprint is started for the given user stories and epics with a particular deadline

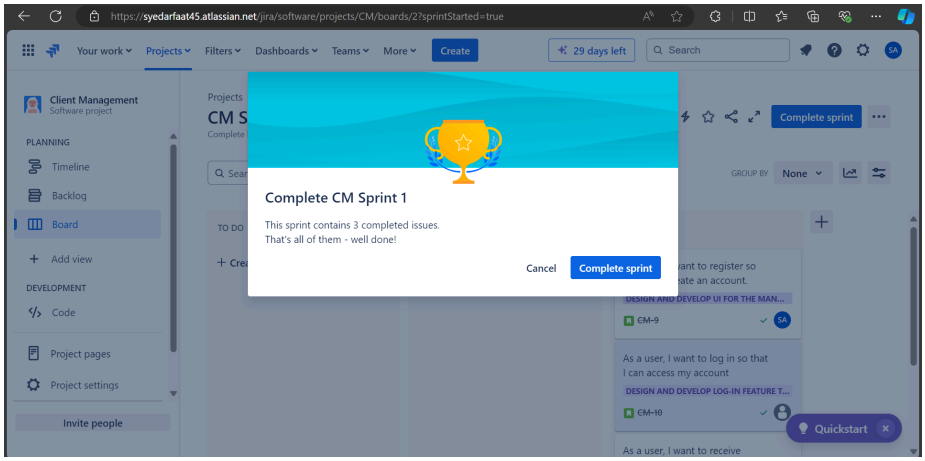




After Completing an epic and user expectation is met, it is then marked as done by the admin/team lead. And when all the epics are completed, the sprint is marked as done.



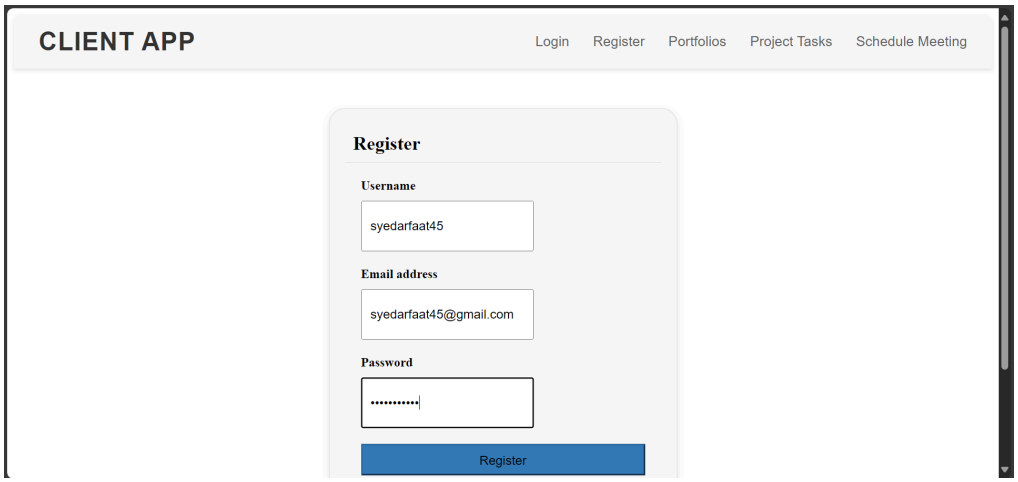




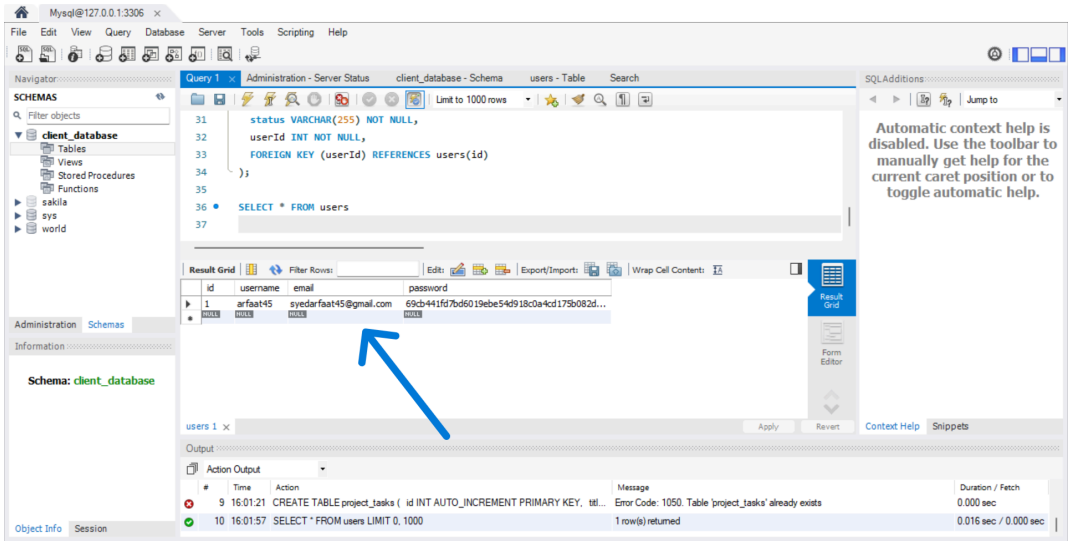
The sprint has been completed for the given requirements. The same is done for the rest of the project.

UI related Screenshots:

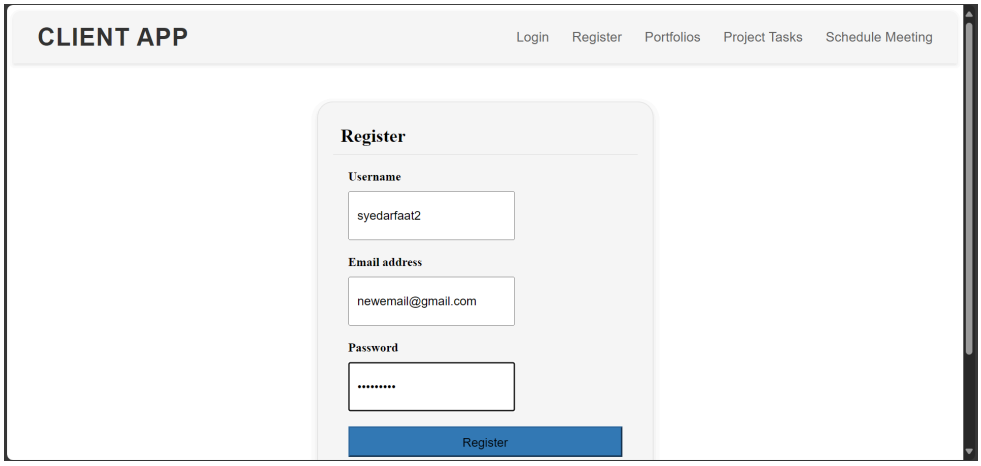
Registering a user:



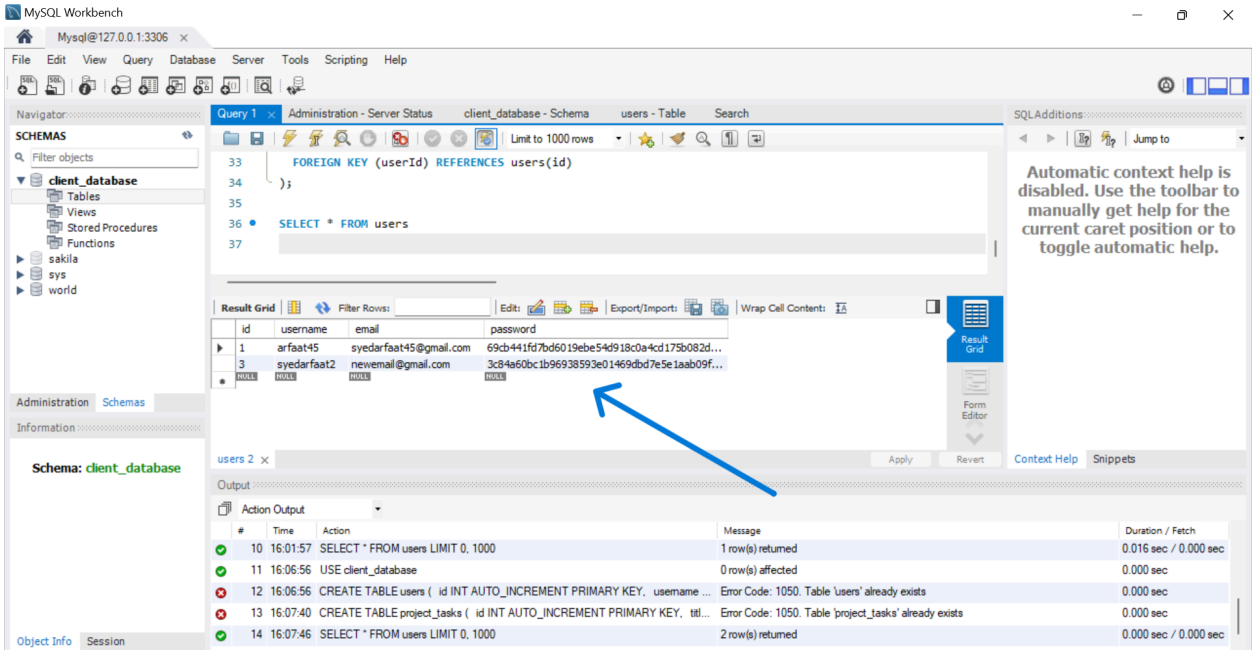
Registering a new user/client. This involves three form inputs that take inputs from the user.
Aftering registration, the user would receive an alert. confirming their registration and then will be redirected to the login page.
Aftering registration, the user's data would be stored in the database created using mysql. The mysql server is running in the background.



Lets Register Another user:

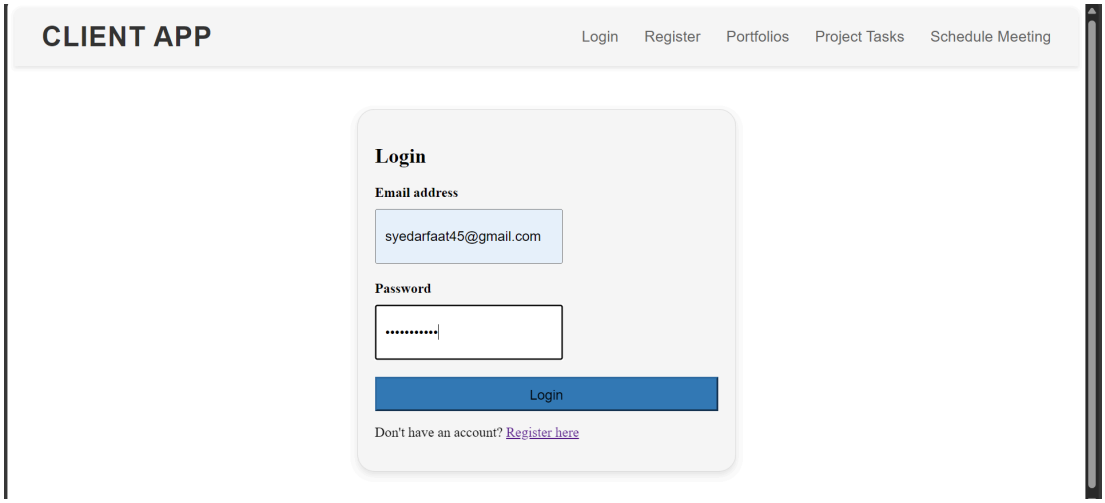


After Register is entered, the data entered in the inputs would be pushed into the mysql database with the help of the backend node.js server running on port 3000.



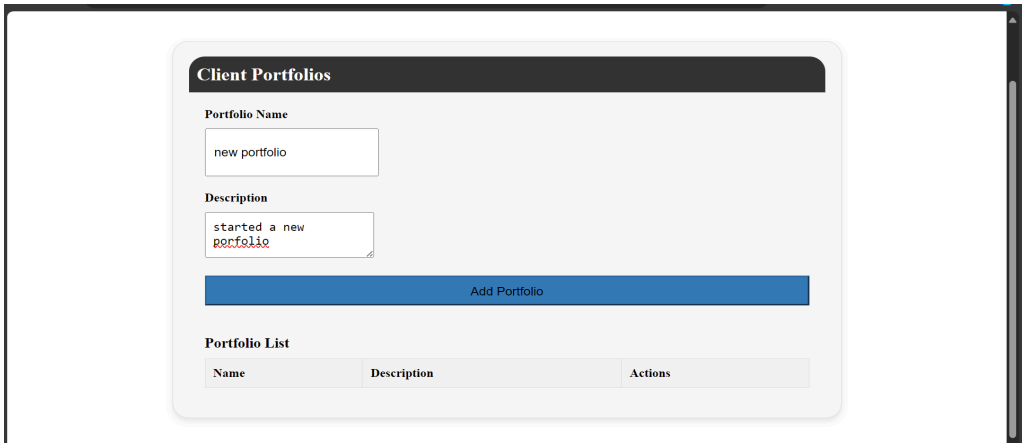
After Registering, the user would be redirected to the login page.

Login Page Related Screenshots:

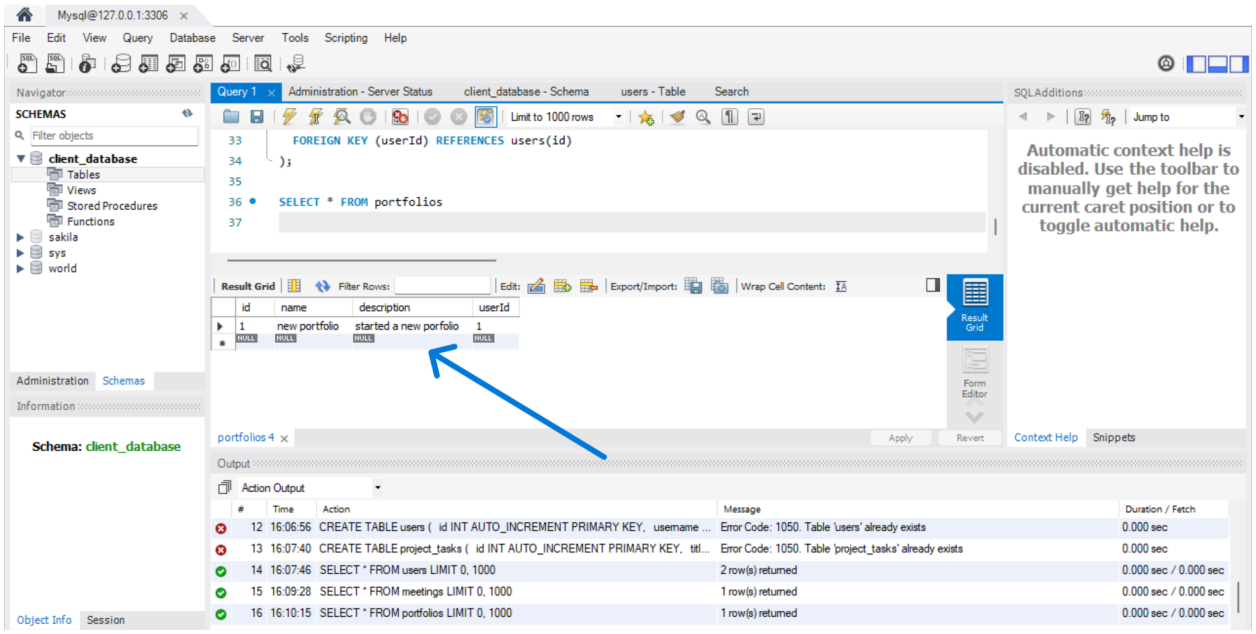


This is the page used to fetch data from the database and validate the user. After Successful login, the user is redirected to the schedule meetings page and an alert pops up mentioning the successful login by the user.

Portfolios Page:



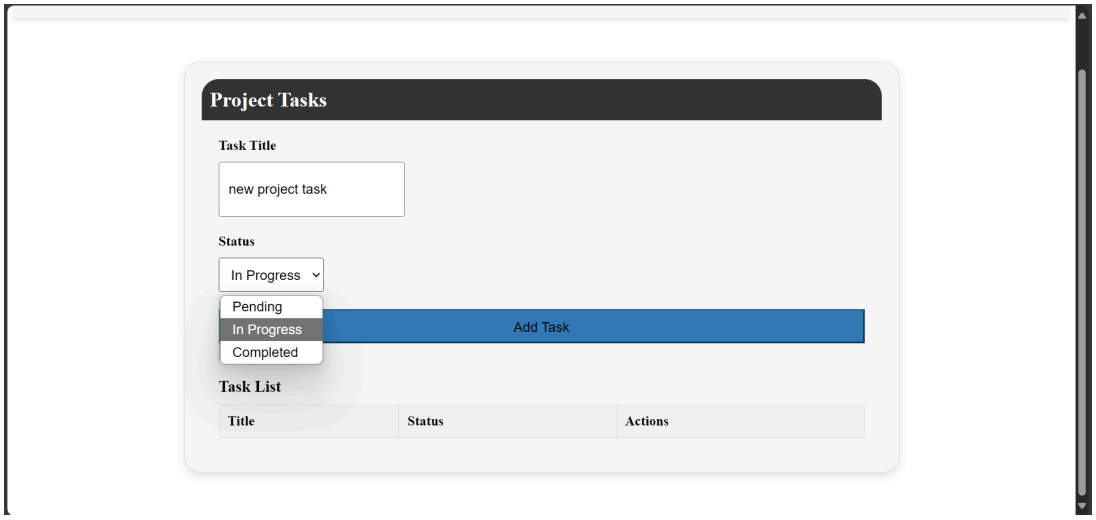
This page allows the user to add new portfolios. After the user clicks the add portfolio button, the portfolio name and description will be added to the mysql database.



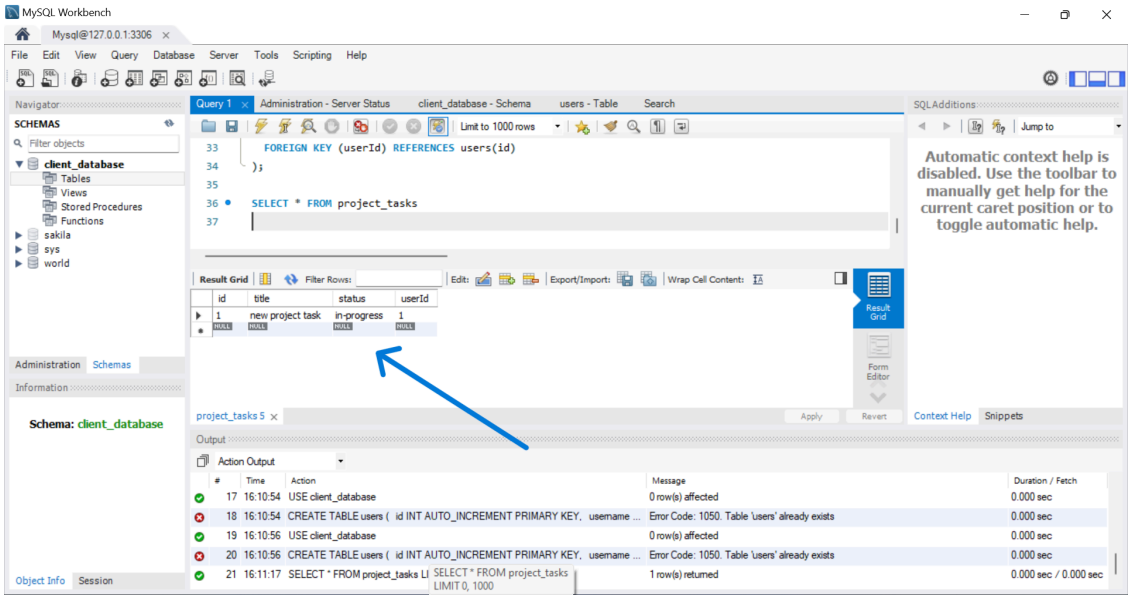
In the screenshot above, we can see the data being entered to the portfolio table in the database and we fetch portfolio details and show it in the UI in the following screenshot.

Project-Tasks Page:

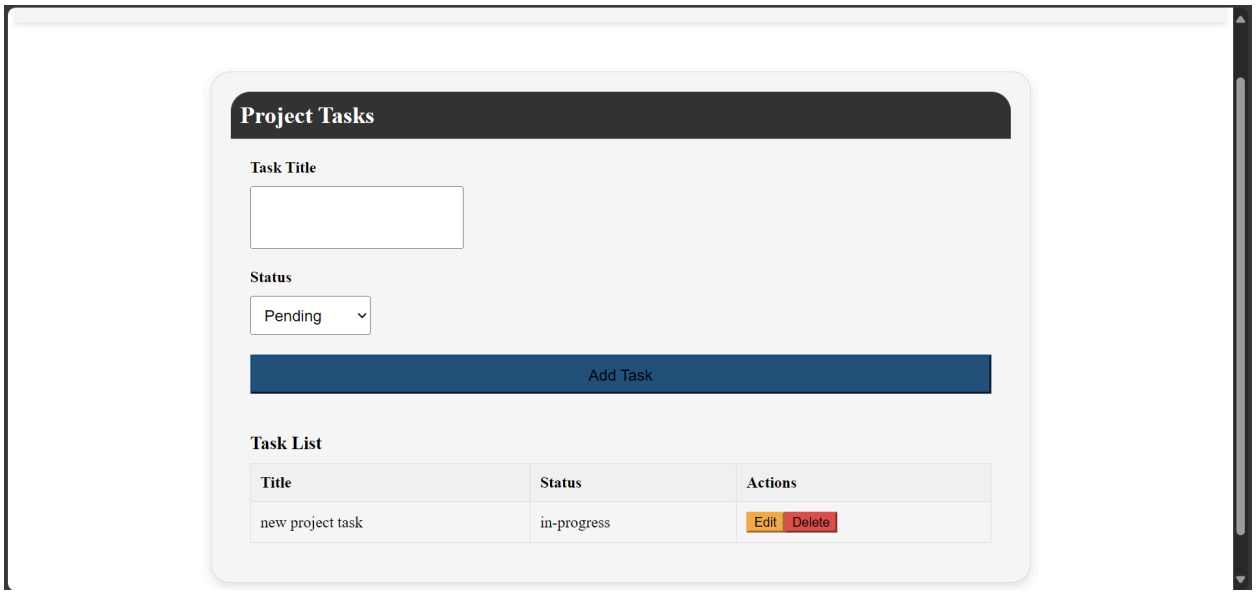
And when navigate to the project tasks page, we can also add project tasks and the data will be stored in the project tasks table in the schema created using mysql workbench.



In the following snapshot, we can see the data being stored in the project tasks table.

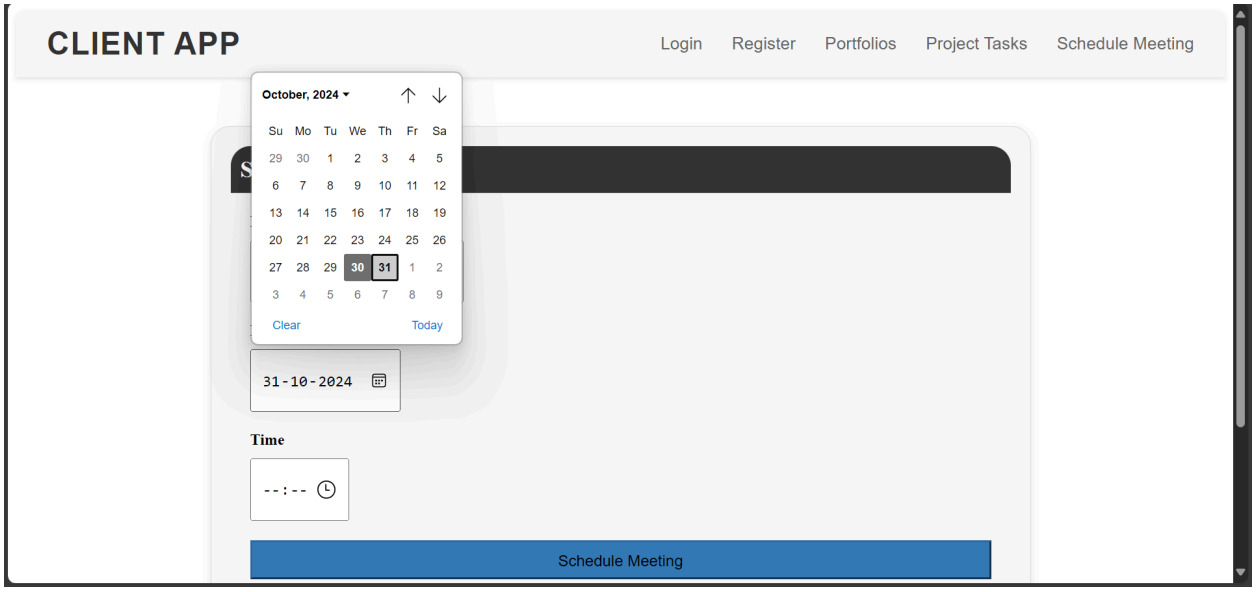


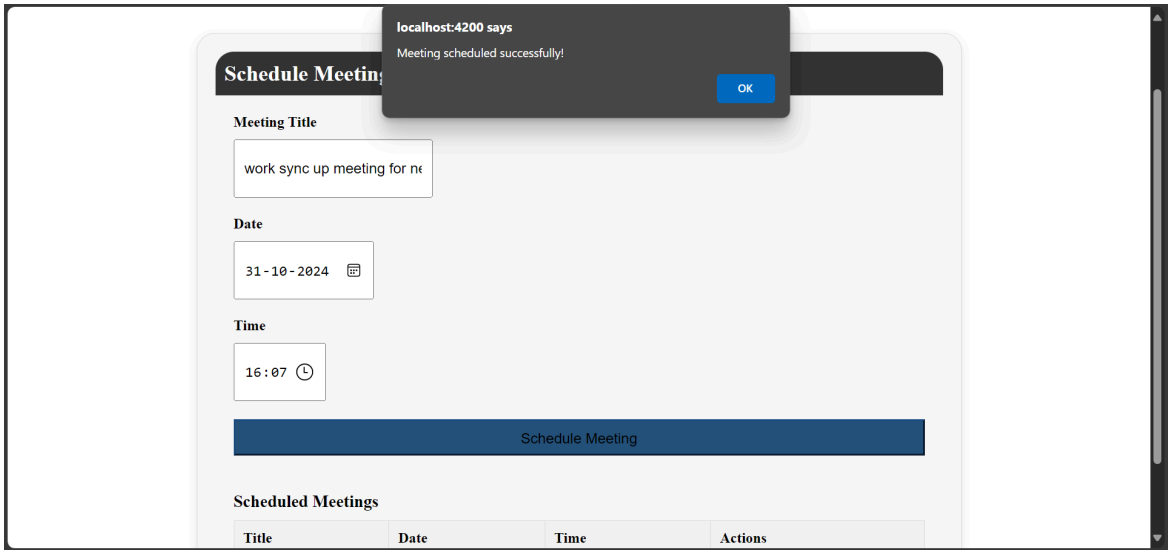
And now we can see the data being displayed on the user dashboard below:



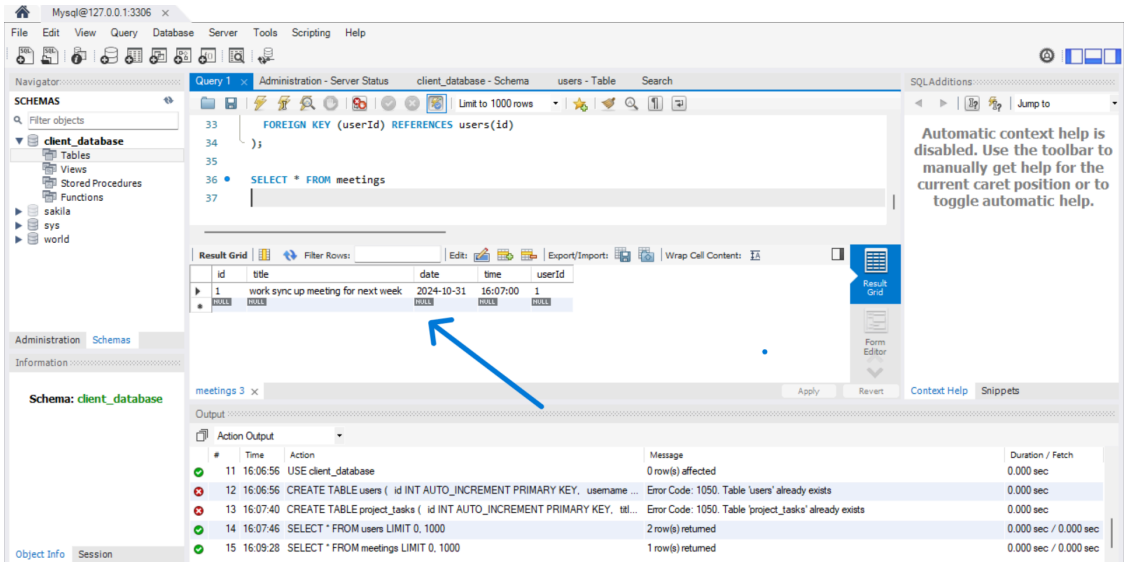
Entering details about various project-tasks assigned.
Status for the Tasks.
An Edit and a Delete Button is there to perform updating and deletion.

Schedule-meeting Page:

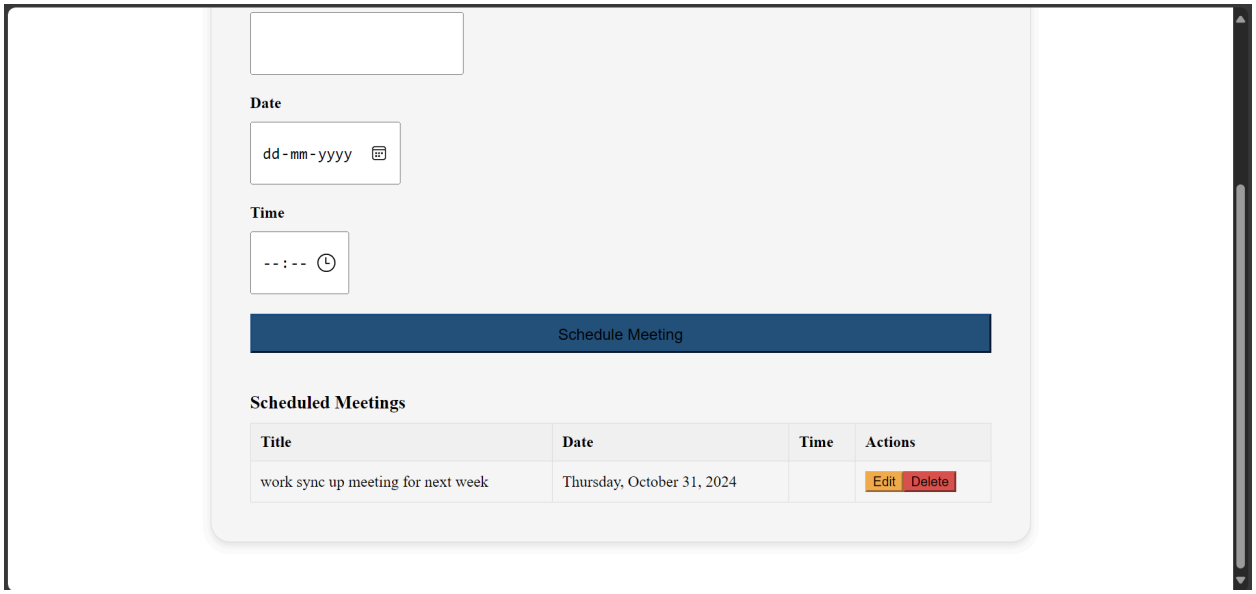




After entering the schedule meeting details the user receives an alert confirming that the details have been updated on the backend.

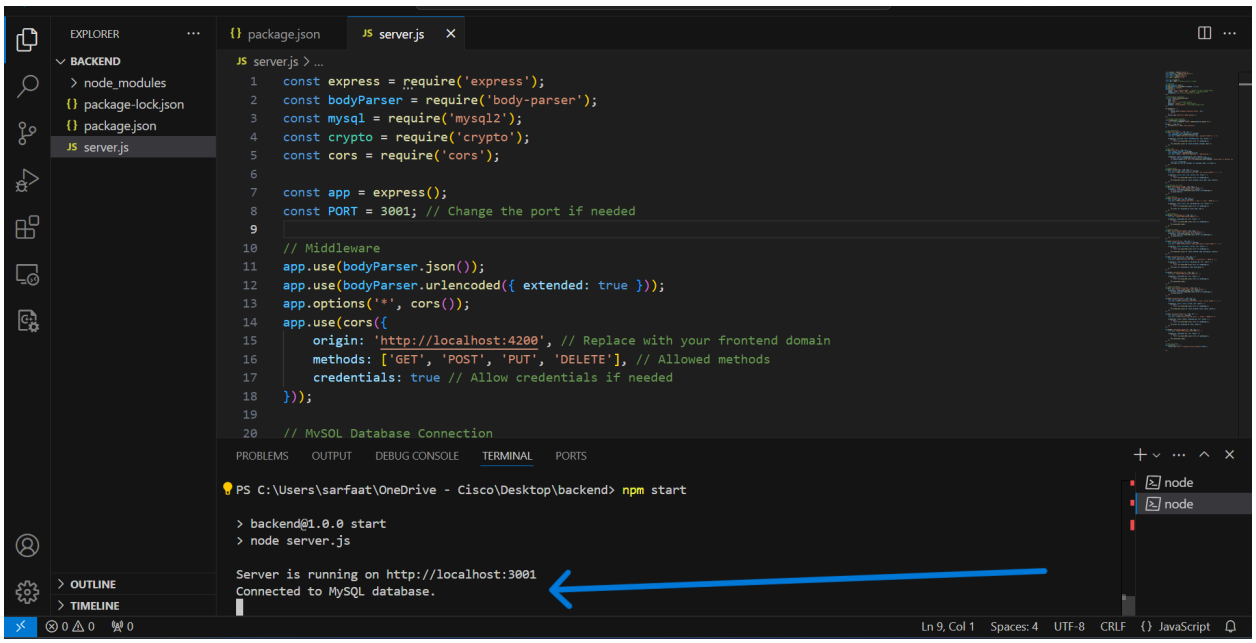


The schedule meting table on the schema/db created on the mysql server running has been updated and this table is fetched and displayed on the user dashboard for schedule meetings page. you can seee this in the Screenshot below:



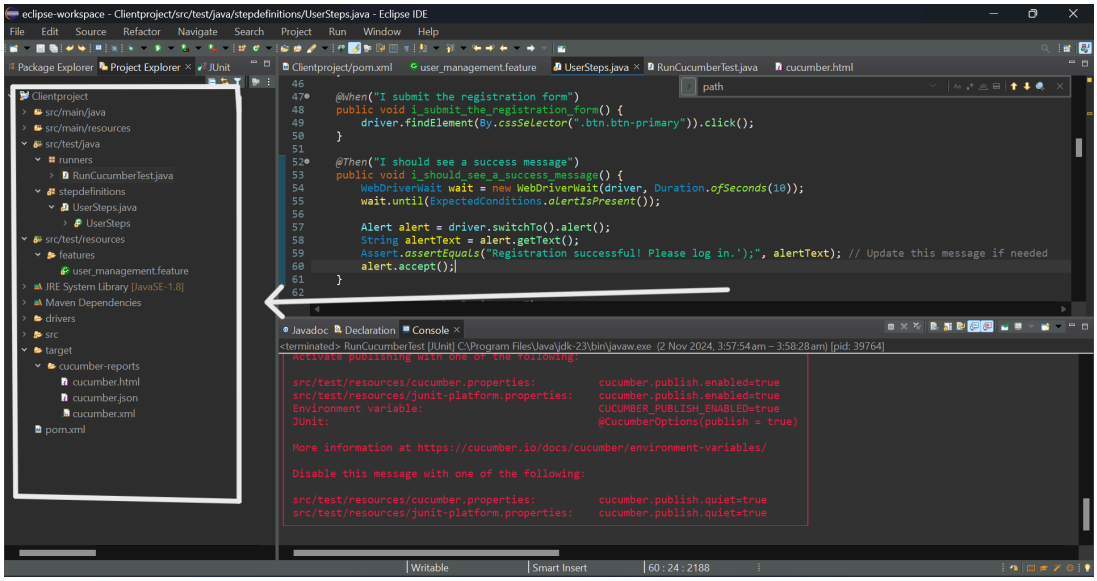
This features provides the user with the following: Entering details about various meetings assigned. An Edit and a Delete Button is there to perform updating and deletion.

Backend Screenshots:

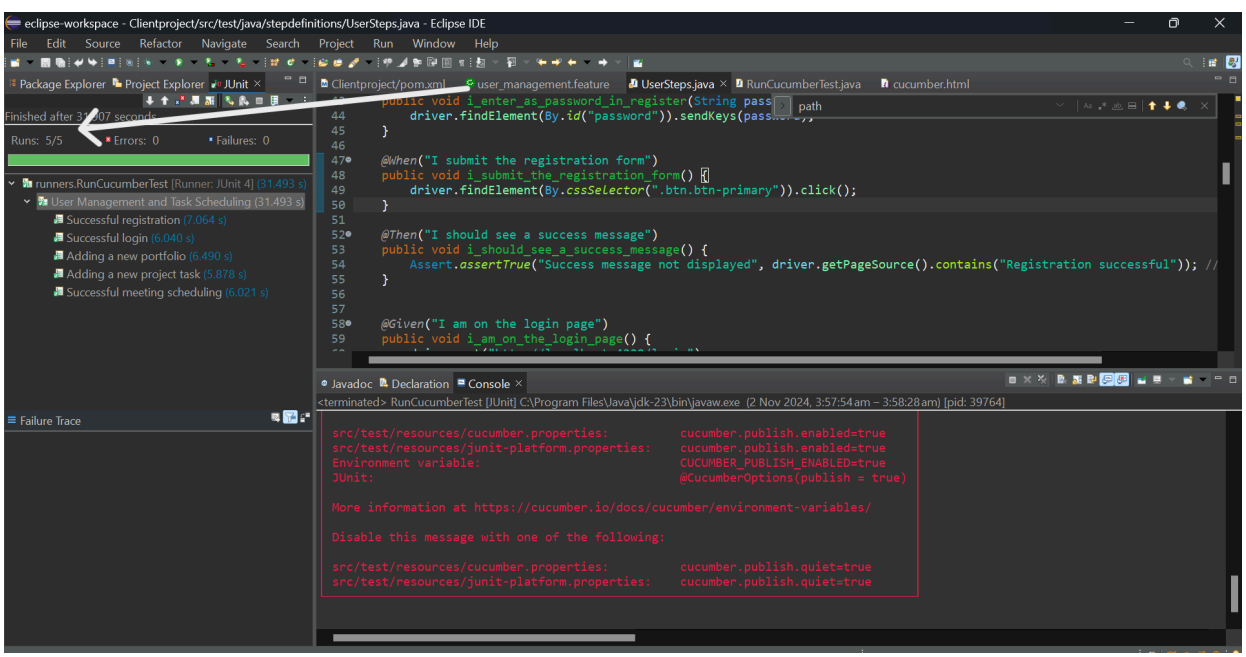


We set up a node.js server that connects the mysql database to the UI created using angular. This node.js backend server listens to the requests made on the ui and fetches data from the database and also does post/update operations to the tables running on the mysql server. It is running on localhost:3000.

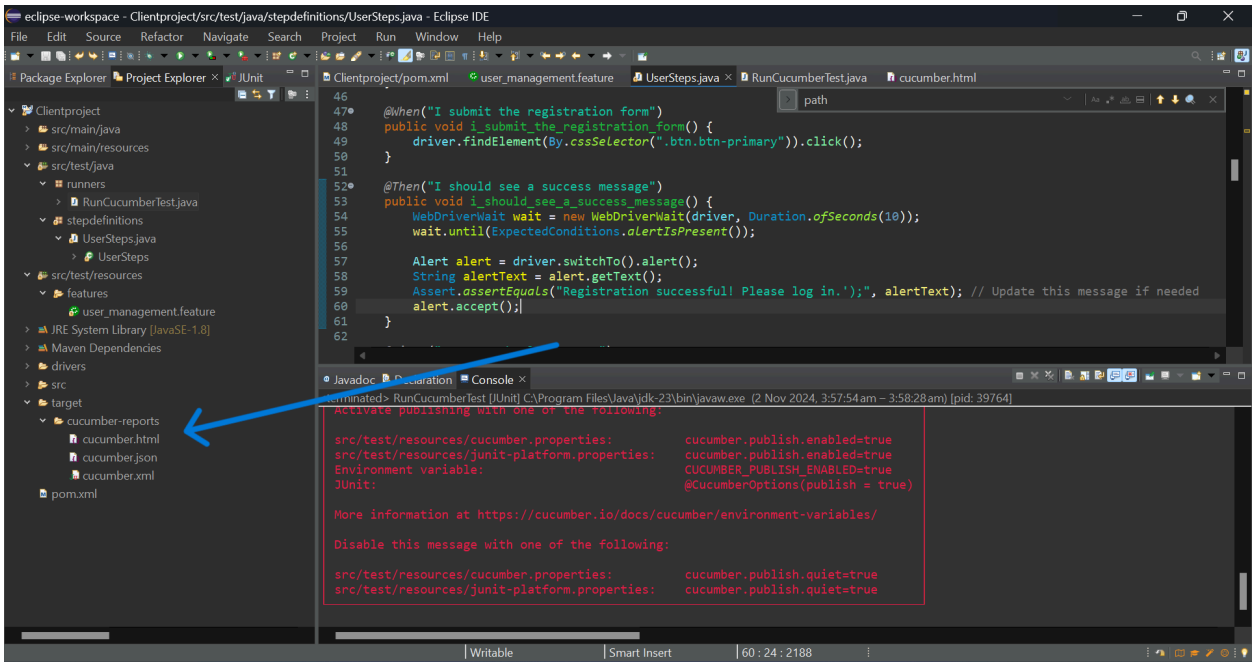
Testing:



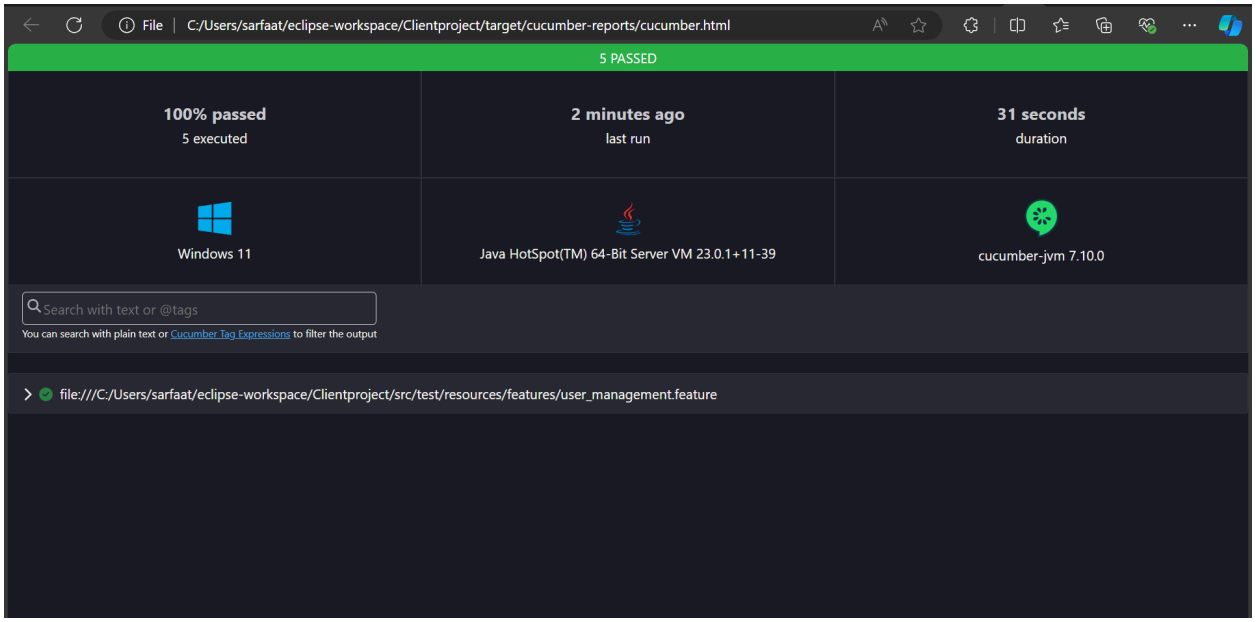
we used eclipse to setup BDD tests using gherkin for client and client management scenarios. This was done using eclipse IDE where step definitions were created and gherkin code was written for multiple scenarios/pages and features were created.

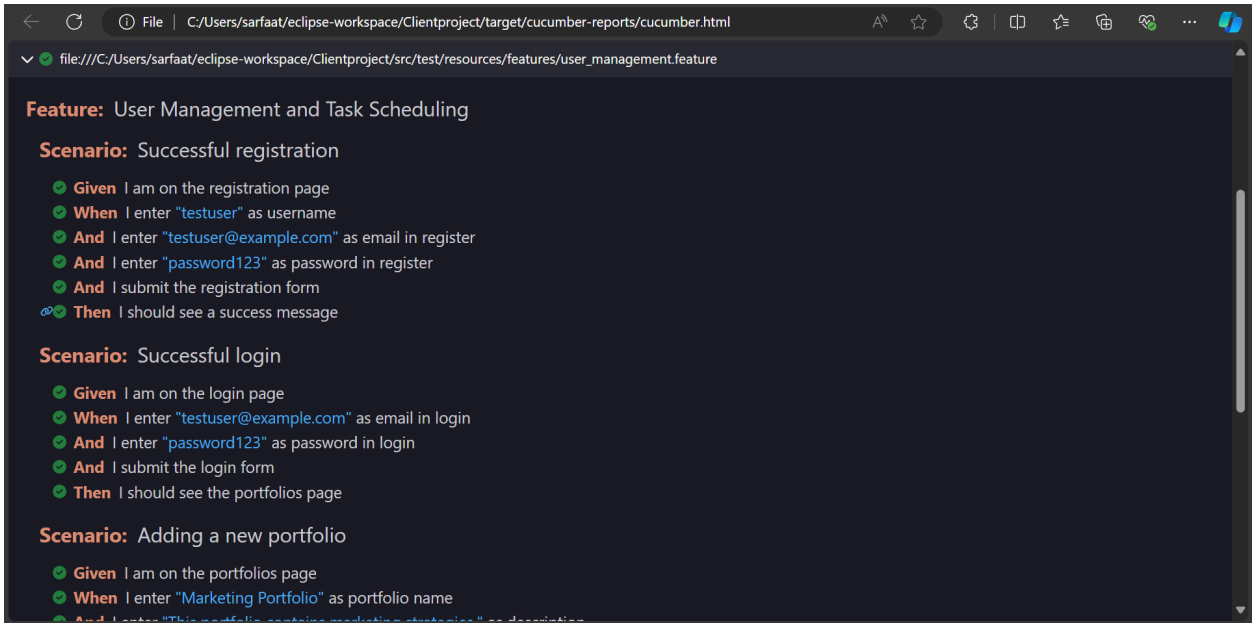


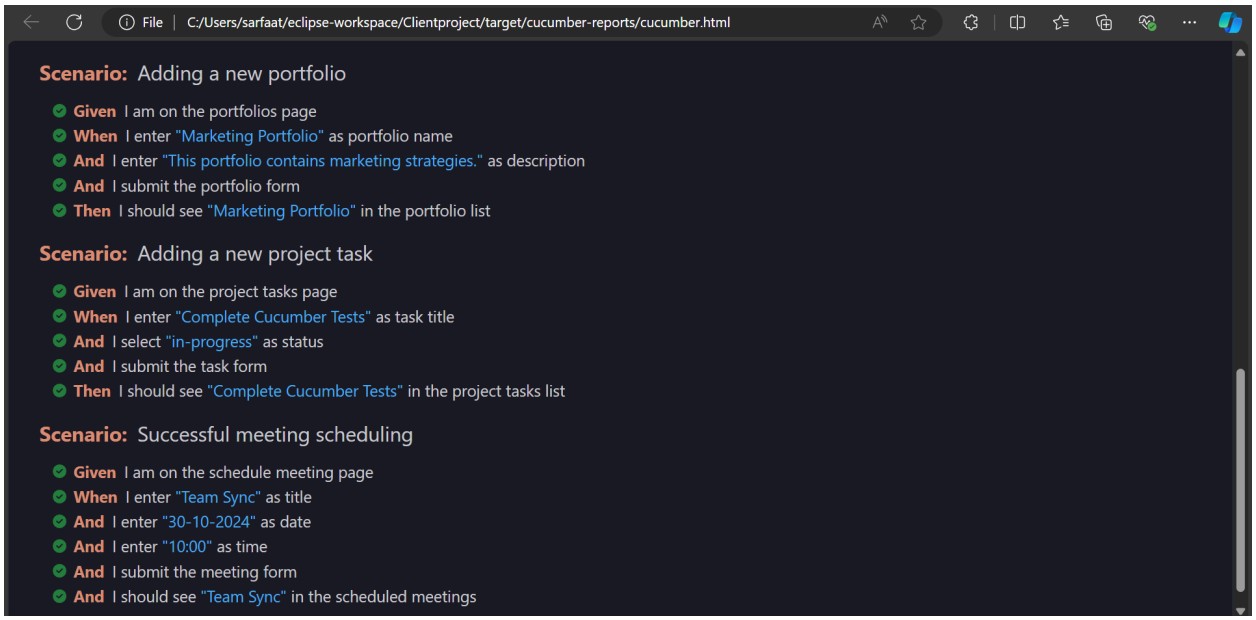
junit cucumber tests were run using selenium with microsoft edge webdriver for automation and then it was tested for each feature. the following are the results which were stored in the target folder:



The following are the results stored in the cucumber.html file:


File | C:/Users/sarfaat/eclipse-workspace/Clientproject/target/cucumber-reports/cucumber.html
5 PASSED
100% passed
5 executed
2 minutes ago
last run
31 seconds
duration
Windows 11
Java HotSpot(TM) 64-Bit Server VM 23.0.1+11-39
cucumber-jvm 7.10.0
Search with text or @tags
You can search with plain text or Cucumber Tag Expressions to filter the output
file:///C:/Users/sarfaat/eclipse-workspace/Clientproject/src/test/resources/features/user_management.feature


File | C:/Users/sarfaat/eclipse-workspace/Clientproject/target/cucumber-reports/cucumber.html
file:///C:/Users/sarfaat/eclipse-workspace/Clientproject/src/test/resources/features/user_management.feature
Feature: User Management and Task Scheduling
Scenario: Successful registration
Given I am on the registration page
When I enter "testuser" as username
And I enter "testuser@example.com" as email in register
And I enter "password123" as password in register
And I submit the registration form
Then I should see a success message
Scenario: Successful login
Given I am on the login page
When I enter "testuser@example.com" as email in login
And I enter "password123" as password in login
And I submit the login form
Then I should see the portfolios page
Scenario: Adding a new portfolio
Given I am on the portfolios page
When I enter "Marketing Portfolio" as portfolio name



The above report shows that the automation tests were successful for the following scenarios of registration page, login page, portfolios page, project tasks page and schedule meetings page. These scenarios were created in the feature file and junit cucumber tests defined inside the userSteps page were successfully run on them and features responded as expected.