P.V.PRAKALYA

192311418

CSA1023-   SOFTWARE ENGINEERING

ASSIGNMENT- 03

Creating a project management board for developing the ARM application for Saveetha University/SIMATS Engineering involves multiple steps that ensure the timely and efficient delivery of the project. We'll use **JIRA** to create the project board and timeline, while also defining the activities to be included in the sprint, the meetings that need to occur before, during, and after the sprint, and other critical aspects of the project management process.

The following answers:

## 1. Setting Up the JIRA Project Board

JIRA is an agile project management tool that allows you to manage tasks in an organized way. You can follow the steps below:

a. Create a New Project in JIRA:

- Go to your JIRA dashboard.

- Create a new project with a Kanban or Scrum board template, depending on your team's process (for this example, we'll assume Scrum for its focus on sprints).

- Name the project "ARM Application Development – Saveetha University/SIMATS Engineering."
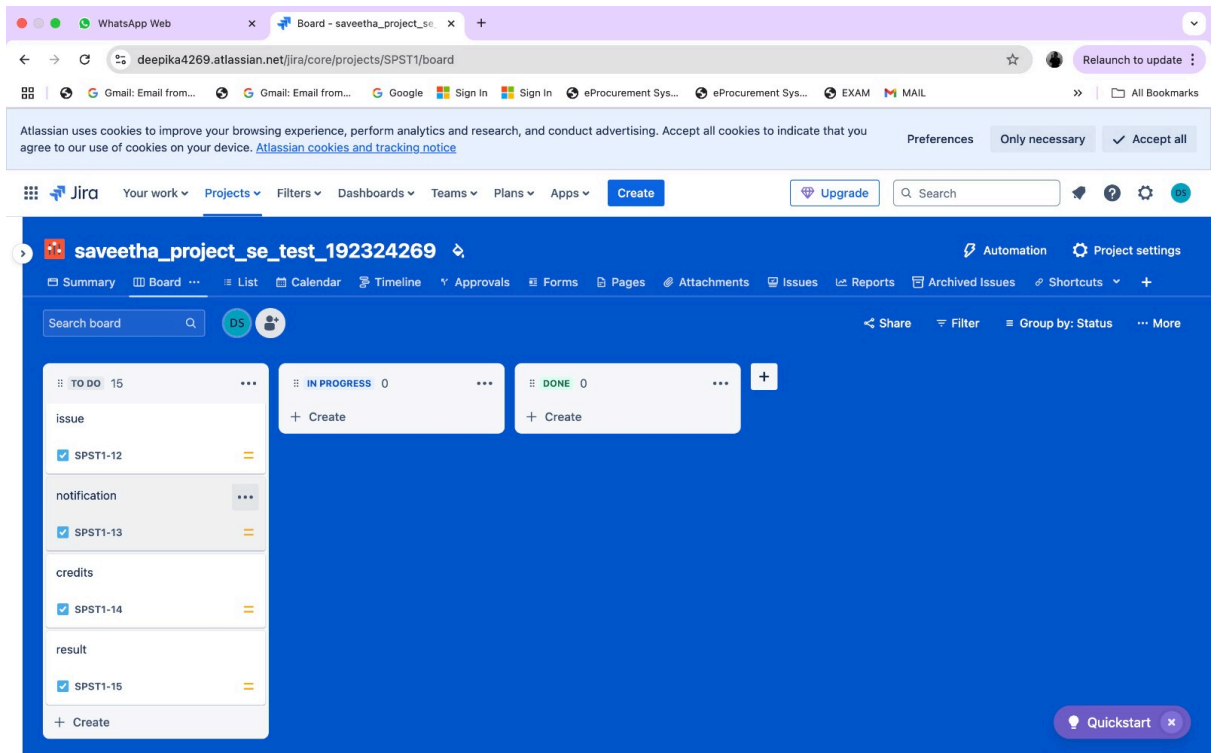
I have give 15 no of tasks in progress board which is in the list of activities that should be given in SIMATS arms.
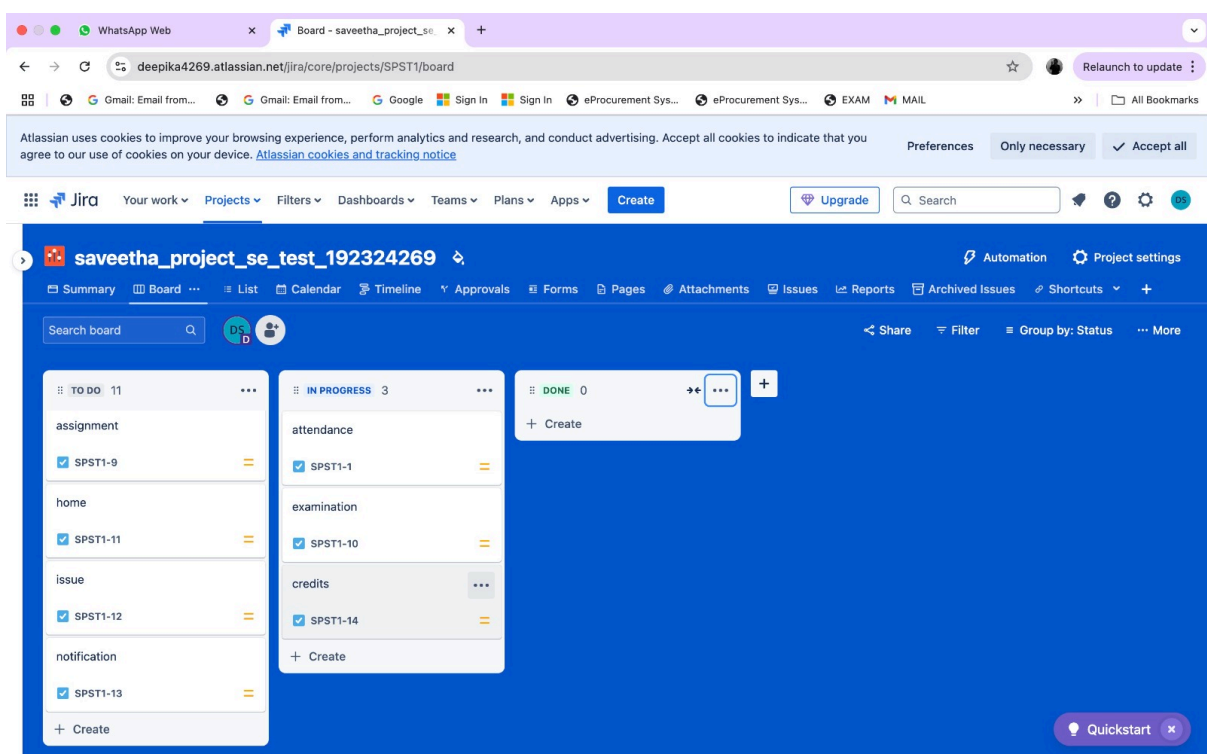
Now this 15 no of tasks is splitted according to the progress to the to
- do list board which as to be done in further

b. Set Up Project Components (Epics):

An Epic is a large body of work that can be broken down into smaller tasks (stories or sub-tasks). You can create Epics for different features of the ARM application. Example Epics might include:

- **Epic 1: User Authentication**

- **Epic 2: ARM Functionality Development**

- **Epic 3: Data Management**

- **Epic 4: User Interface (UI) Design**

- **Epic 5: Integration and Testing**

c. Create Tasks and Stories:

For each Epic, create corresponding tasks. If you have 15 tasks, they will likely be assigned across the Epics. For instance:

- **Epic 1: User Authentication**

    - Task 1: Design login page

    - Task 2: Implement login functionality

   - Task 3: Integrate with database for authentication

Epic 2: ARM Functionality Development

- o Task 4: Develop core ARM algorithms

- o Task 5: Test ARM functionality in a sandbox environment

d. Add Sub-tasks (if necessary):

If tasks are large or complex, break them into sub-tasks. For example, "Test ARM functionality" could have sub-tasks like:

- Sub-task 1: Unit test ARM functionality

- Sub-task 2: Integration testing with other modules

- Sub-task 3: Performance testing.
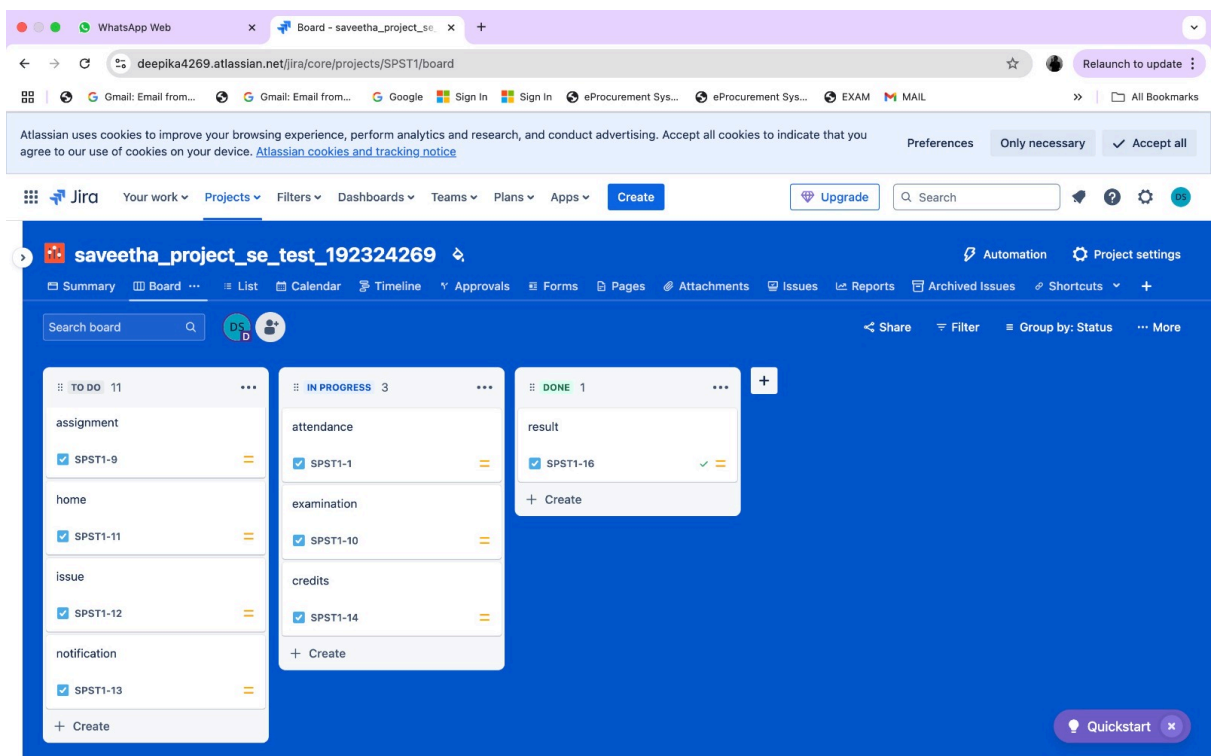
And also when the activity is done .. we can move to done list board.

e. Assign Tasks and Set Priorities:

Ensure tasks are assigned to specific team members and given priorities (High, Medium, Low) depending on their urgency.

---

2. Timeline Board

The timeline can be visualized using JIRA's Sprint Planning feature or Gantt chart tools if needed. Below is a sample timeline for 4 sprints, assuming the project is planned for around a month with each sprint lasting 2 weeks:

Sprint	Start Date	End Date	Main Tasks
Sprint 1	10-Feb-2025	23-Feb-2025	1. Task 1: Design login page2. Task 2: Implement login functionality

Sprint	Start Date	End Date	Main Tasks
Sprint 2	24-Feb-2025	07-Mar-2025	1. Task 3: Integrate with DB2. Task 4: Develop core ARM algorithms
Sprint 3	08-Mar-2025	21-Mar-2025	1. Task 5: Test ARM functionality2. Task 6: Design UI components
Sprint 4	22-Mar-2025	04-Apr-2025	1. Task 7: Final testing2. Task 8: Deployment to staging

Each sprint will be planned to include the necessary tasks and subtasks for completion, following JIRA's task tracking system.

---

3. Sprint Backlog & Ordered List of Activities

The ordered list of activities (based on priority) should be finalized during Sprint Planning meetings. It will include the following:

Sprint Backlog (Ordered List of Activities):

1. **Design login page**

2. **Implement login functionality**

3. **Integrate authentication with database**

4. **Develop core ARM algorithms**

5. **Unit test ARM functionality**

6. **Integration testing with other modules**

7. **Performance testing of ARM algorithms**

8. **Design UI components for ARM application**

9. **User interface testing**

10. **Final system integration and deployment testing**

11. **Set up staging environment**

12. **Deploy to production**

13. **Final user acceptance testing**

14. **Post-deployment monitoring and bug fixes**

15. **Release project documentation and support**