Team number:

Feinan Guo, Jiahui Huang, Ruxin Ma, Shiyu Zhao, Xuran Chen

Project Title:

Project LS1

AI-Driven Warehouse Packing and Visualization Solutions

Milestone 1	Activities		Planned Outputs	Achieved Outputs
Restate the milestone from your Draft plan.	Restate the key activities from your draft plan.		Restate the planned outputs from your draft work plan.	Outline the actual outputs compared to what was projected (or type "same as planned")
Establish project fundamentals (through planning, requirements gathering, technical design, and basic UI development)	Project Plan	Create project timeline	Gantt chart with key milestones	Delivered comprehensive Sprint Plan (timeline) Week 3. Successfully broke down activities and outputs Milestone 1- focused schedule. [ref1, Appendix-Section1]
		Establish team structure	Team roles and responsibilities document	Completed the draft roles and responsibilities document in Week 5 assignment. Finalized with clear assignment of each role to specific team members in Week 6. [ref2, Appendix-Section2]
		Develop a communication strategy	Communication plan	Same as planned [ref3, GitHub Repository/main]
	Requirements Analysis	Gather client requirements	Product Requirements Document (PRD)	Same as planned [ref3, GitHub Repository/main]
		Map user flows	User journey diagram	Same as planned [ref3, GitHub Repository/main]
		Create user stories	User story with acceptance criteria	Same as planned [ref4, Appendix-Section3]
	Technical plan	Select technology stack	Technology stack documentation	Same as planned [ref3, GitHub Repository/main]
		Design system architecture	System architecture diagram	Same as planned [ref3, GitHub Repository/main]
		Define API endpoints	API design document	Same as planned [ref3, GitHub Repository/main]
		Create database schema	Database schema design	Same as planned [ref3, GitHub Repository/main]
	Development	Set up dev environment and build the main application layout	Basic UI framework implementation	Same as planned [ref5, GitHub Repository/dev-frontend; ref6, Appendix-Section4]
		Develop login/sign-up interface	Functional login/registration UI	Same as planned [ref6, Appendix-Section4]
		Define API contracts	API specification document	6 APIs in total: 1 API completed; 4 APIs reviewing and expected week 8 completed; 1 APl planned week 9. [ref3, GitHub Repository/main]
		Configure database connections	Working database connection	Database setup completed with the User table created; remaining tables for items, tasks, containers, and related data will be added upon API implementation completion.
	Testing Plan	Identify key features for testing	Feature testing checklist	Same as planned [ref3, GitHub Repository/main]

GitHub Repository: https://github.cs.adelaide.edu.au/MCI-Project-2025/LS1

Team reflection on progress	Provide some comments below regarding the completion of this milestone specifically around: 1. How is the project progressing? 2. Are there any differences between projected and actual outputs/outcomes?

1. Project Progress

All project phases are **progressing on schedule**, with significant advancements in planning, requirements analysis, and both front-end and back-end development.

Project Planning

We implemented a structured a *sprint plan* in Week 3 that identified key activities. Then we drafted a detailed *Gantt chart* with detailed activates in Week 5 (see Appendix 1, Figures 1-2). This plan helped us to be able to stay focused and monitor progress effectively on a weekly basis. In addition, we planned weekly team meetings on Fridays, and bi-weekly client meetings to ensure synchronization of information and transparency of progress. In weeks 5-6, we formalized our team structure with clear divisions for front-end, back-end, and project management responsibilities. This organization enhances team cooperation, task authorization and overall responsibility.

Requirements Analysis

Our requirements phase produced a comprehensive *Product Requirements Document (PRD*) that thoroughly documents customer requirements. We also designed user *journey diagrams* to clearly illustrate the core interactions for both manager and worker roles. Additionally, we created *user stories* with clear acceptance criteria and documented them in the backlog list. These efforts provided clear direction and basis for the design and implementation phases.

Technical Development

The **front-end** development has completed several critical components, including the user login and registration interfaces, role-based access control logic, the main interface layout for both manager and worker dashboard and implement the 3D/2D visualization using three.js (see Appendix 4, Figures 5-9), which has laid a good foundation for the subsequent task operation functions. At the same time, team members proactively developed *feature testing checklists* during UI implementation, which will serve as the framework for our comprehensive testing strategy in the next sprint. We also completed the deployment of Docker containers and Nginx service configuration, and wrote a deployment document, initially establishing a deployment process applicable to this project.

On the **back-end**, we implemented the Flask framework in the dev-backend branch and began building out the main files within the */backend* directory. The *login API* is now fully operational and documented. We have also successfully connected to the database and set up and tested the *User table*. We are also halfway to completing the Docker and virtual environment and making for easier testing and deployment. All development efforts remain on target according to the project timeline, with integration tasks scheduled for completion by the end of the current sprint.

2. Differences

Project Planning Adjustments

Project planning is proceeding on schedule for most tasks with only one notable change. We postponed team **role assignments** from Week 5 to Week 6 due to unclear

requirements (see Appendix 2, Figure 3). This delay allowed us to ultimately assign roles based on team members' technical backgrounds, which improved our team structure and resulted in better execution efficiency and cooperation.

Back-End Development Gaps

The key difference in back-end development is the number of completed **APIs**. We planned six key APIs to be completed and reviewed by this sprint, but only completed one (the *login API*). The other five APIs are still in progress or under review. Regarding **database** development, we have only completed the User table. The remaining tables for items, tasks, containers, and related data are pending and will be added upon completion of their respective API implementations. Despite these gaps, the core back-end structure is now properly organized, and we expect to complete all remaining APIs and database tables by the end of the sprint.

Team reflection on managing problems	Have you encountered any problems to date? If so, how have you managed them?

Task Allocation Challenges

Initially, we faced insufficient team interaction resulting in uneven task distribution. Two members were overcontributing while others showed limited engagement. After supervisor intervention, all members accepted responsibility for specific modules. When workload imbalances became apparent, we implemented self-assessment of task difficulty and time estimates. Team members with heavier workloads requested help, while those with lighter workloads provided support.

Task Progress Issues

Despite exceeding initial overall progress expectations, several key deliverables were delayed. The Docker platform was completed two weeks behind schedule, and the API specification document missed its agreed deadline. To resolve these issues, we implemented GitHub backlog Kanban boards to visualize workflow bottlenecks (see Appendix 5, Figure 10) and increased communication in our development chat. As a result, we successfully achieved all major milestones including technical design, platform layout, and key user interfaces.

Version Control Problems

We addressed specific workflow inconsistencies during development:

1. A team member bypassed the pull request workflow and merged directly to the main branch during Worker Dashboard development, increasing potential conflicts and complicating code tracking. We resolved this by implementing mandatory PR-based submission with formal code reviews (see Appendix 6, Figures 11-12).
2. During registration page development, a team member did not reference existing login components and requirements, creating inconsistencies. The team lead identified these issues during PR review, provided feedback, and the member reorganized the page to maintain unified style and functionality.

Overall, effective communication and collaboration have enabled the team to overcome these challenges while maintaining better-than-expected progress.

Supervisor assessment	Please, rate your team (1) effort, (2) project progress and (3) their self-reflection for milestone 1 Rating scale 1-10 as per standard marking scheme, ie 5 is a Pass and 7 is a credit. Add some comments to explain your rating
Effort: Progress: Reflection:	

Appendix

Contents

Section 1 Project Plan *7*

Section 2 Team Organization *9*

Section 3 Users' Story *10*

Section 4 User Interfaces of the Platform *11*

Section 5 Project Backlog *16*

Section 6 GitHub Pull Request *17*

Section 1 Project Plan

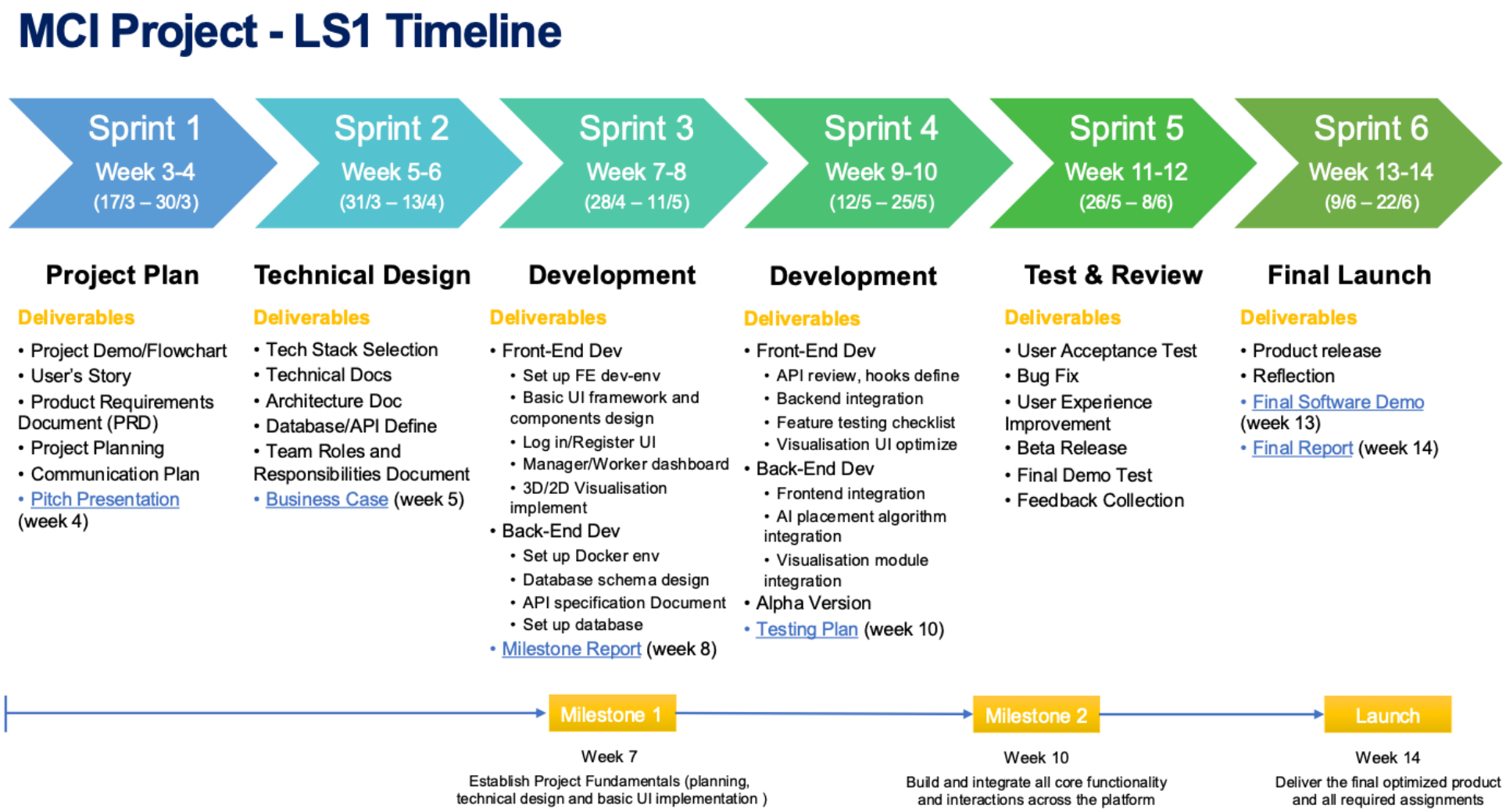

Figure 1 MCI Project - LS1 Timeline

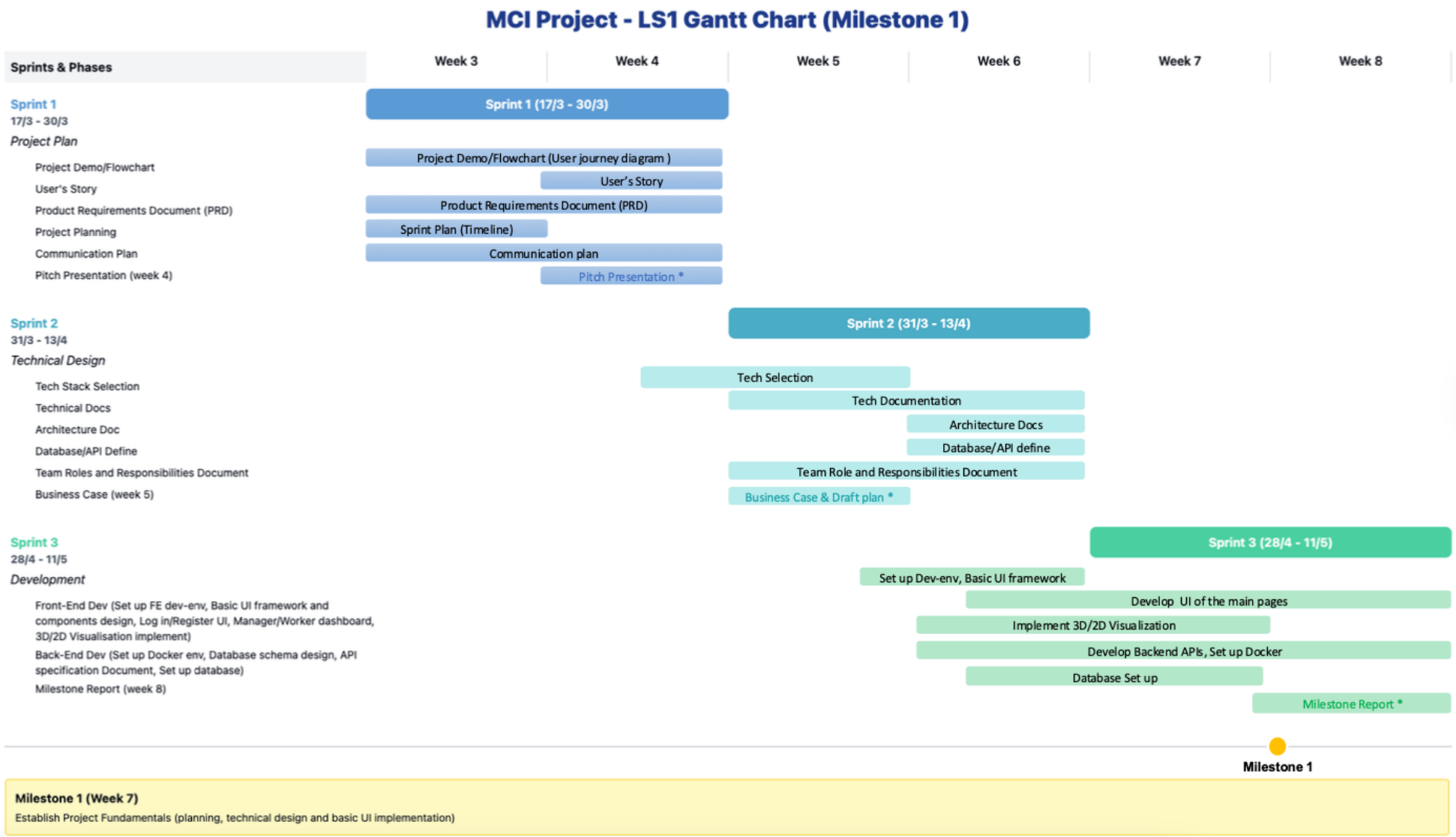

Figure 2 MCI Project - LS1 Gantt Chart (Milestone 1)

Section 2 Team Organization

Team Organization V1.0

Internal Roles

Role	Responsibilities	Assignment Approach
Project Coordinator	Meeting facilitation, risk management, progress reporting, team coordination	Shared/Rotating weekly among all members
Requirements Analyst	Requirement gathering, feature specification, user story and acceptance criteria definition	Collaborative effort by all team members
Full-Stack Developer	Frontend UI implementation, backend API development, database design, integration testing	All team members contribute based on current project needs
3D/2D Visualization Specialist	Implementation of warehouse visualization components, algorithm output rendering, and interactive display features	Team members with relevant expertise

External Stakeholders

Role	Responsibilities	Entity
Algorithm Specialist	Development of optimization algorithms, performance tuning, and integration support	External Developer
Supervisor	Technical guidance, project evaluation, and client communication facilitation	Course Instructor
Client	Business requirements definition, feedback on deliverables, final acceptance	Client Representative

Team Organization V2.0

Internal Roles

Role	Responsibilities	Assigned Student
Project Coordinator	Meeting facilitation, risk management, progress reporting, team coordination	Shared/Rotating weekly among all members
Requirements Analyst	Requirement gathering, feature specification, user story and acceptance criteria definition	Collaborative effort by all team members
Front End Developer	UI implementation, backend integration, user experience design	Ruxin Ma (Lead) Shiyu Zhao Jiahui Huang
Back End Developer	API development, database design, integration with frontend/algorithms/visualization	Feinan Guo (Lead) Xuran Chen Jiabao Ye
3D/2D Visualization Specialist	Warehouse visualization components, algorithm output rendering, interactive display features	Jiahui Huang

External Stakeholders

Role	Responsibilities	Person
Algorithm Specialist	Optimization algorithm development, backend integration support	Jiabao Ye
Supervisor	Technical guidance, project evaluation, and client communication facilitation	Lia Song (Course Instructor)
Client	Business requirements definition, feedback on deliverables, final acceptance	Dr. Liu (Client Representative)

Figure 3 Team Organization

Section 3 Users' Story

https://github.cs.adelaide.edu.au/orgs/MCI-Project-2025/projects/21/views/1

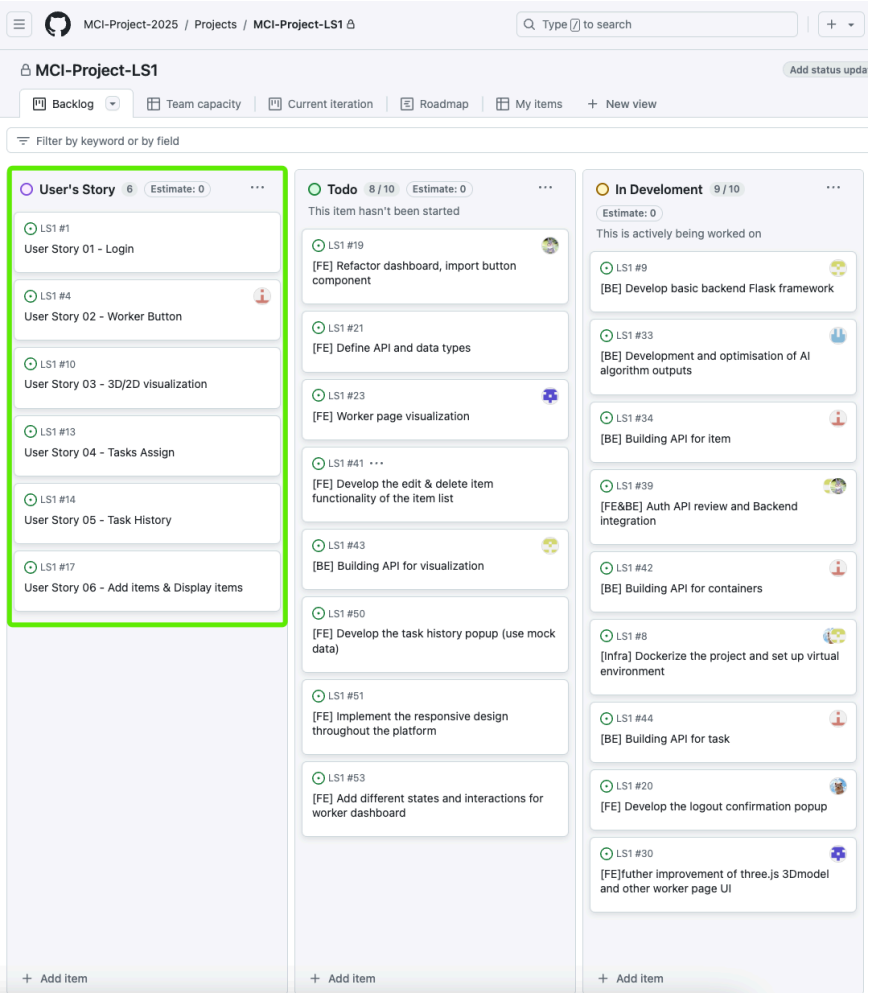

Figure 4 Screenshot of User's story

Section 4 User Interfaces of the Platform

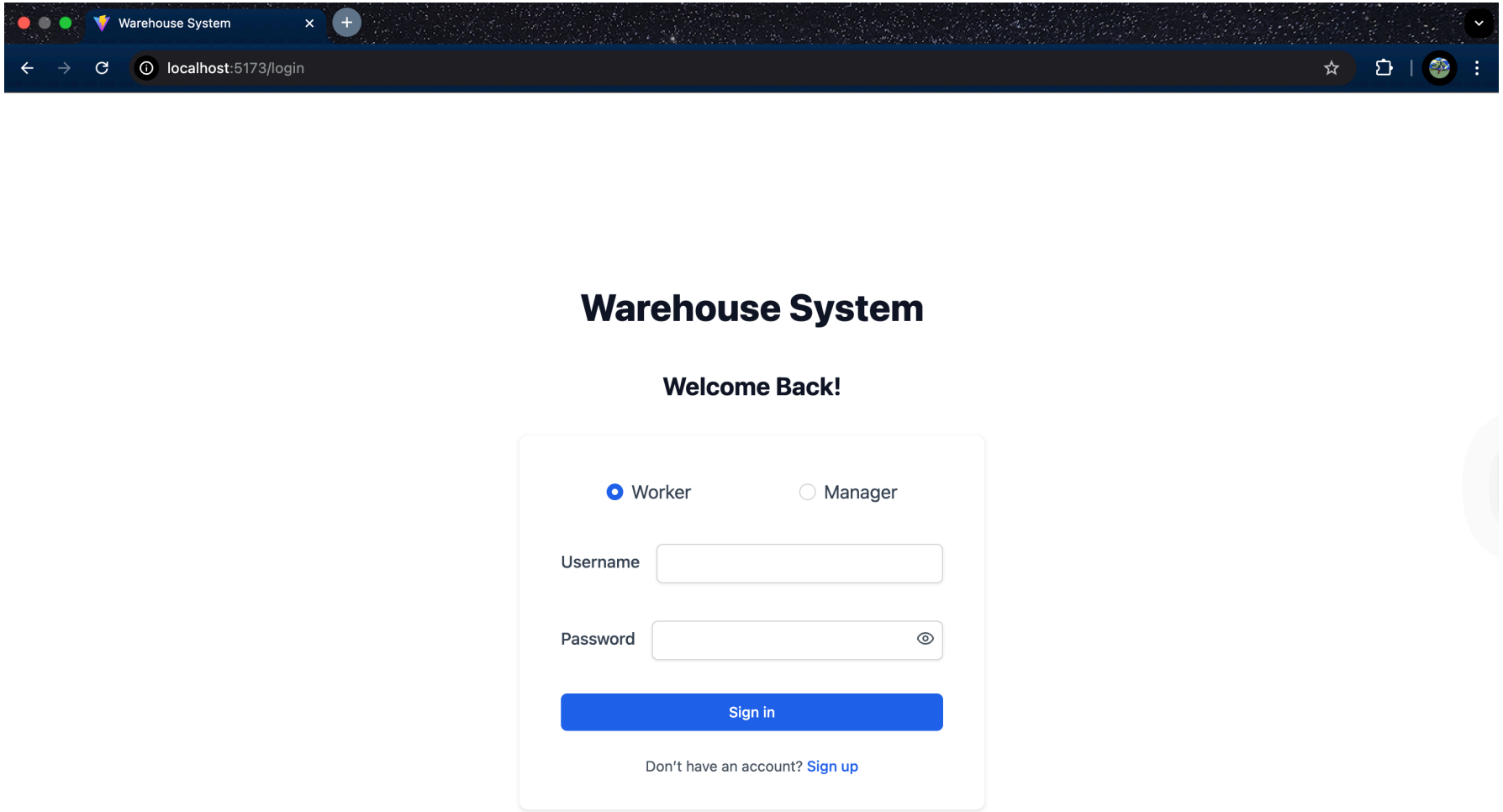

Figure 5 Login Page User Interface

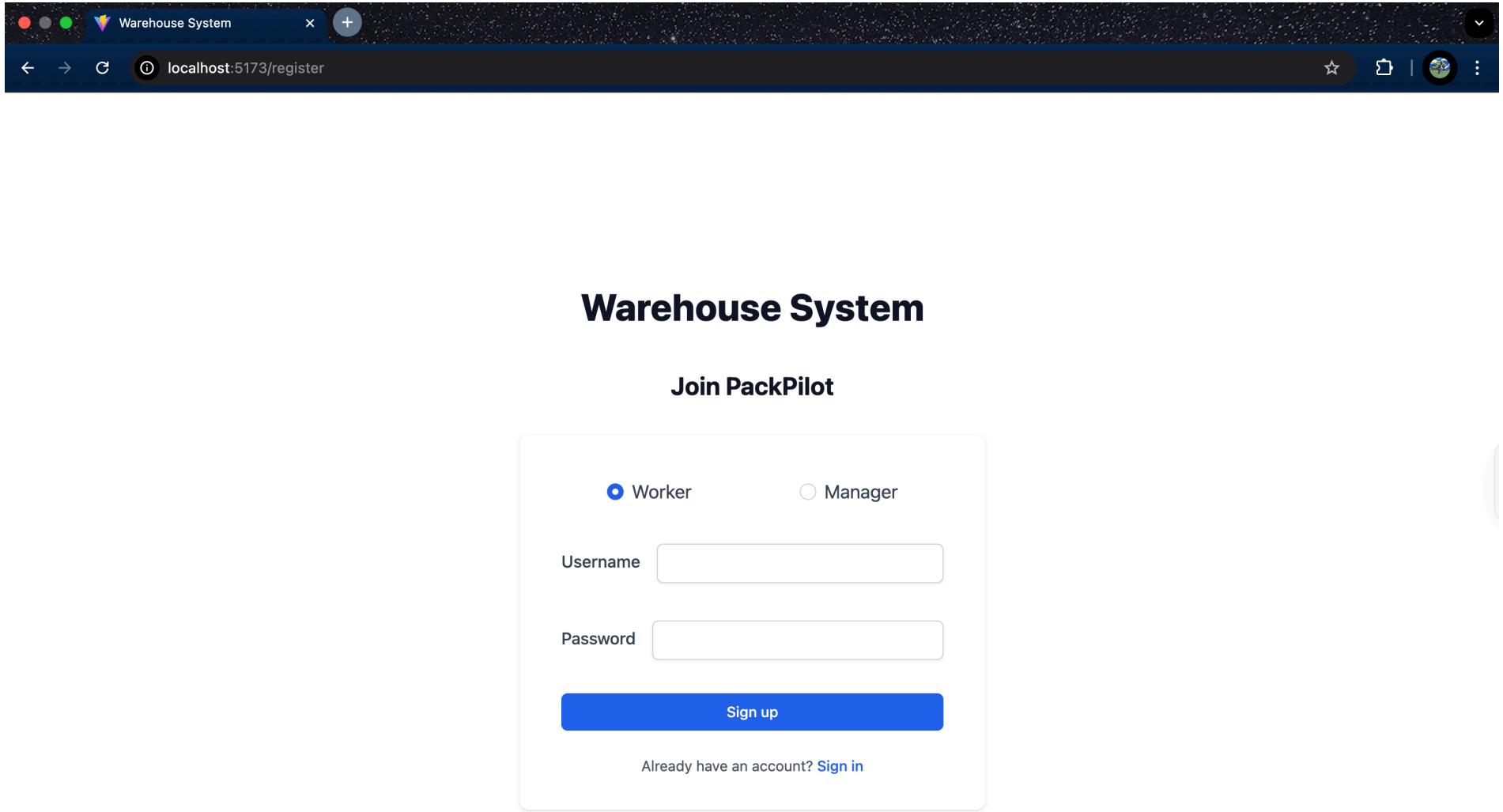

Figure 6 Register Page User Interface

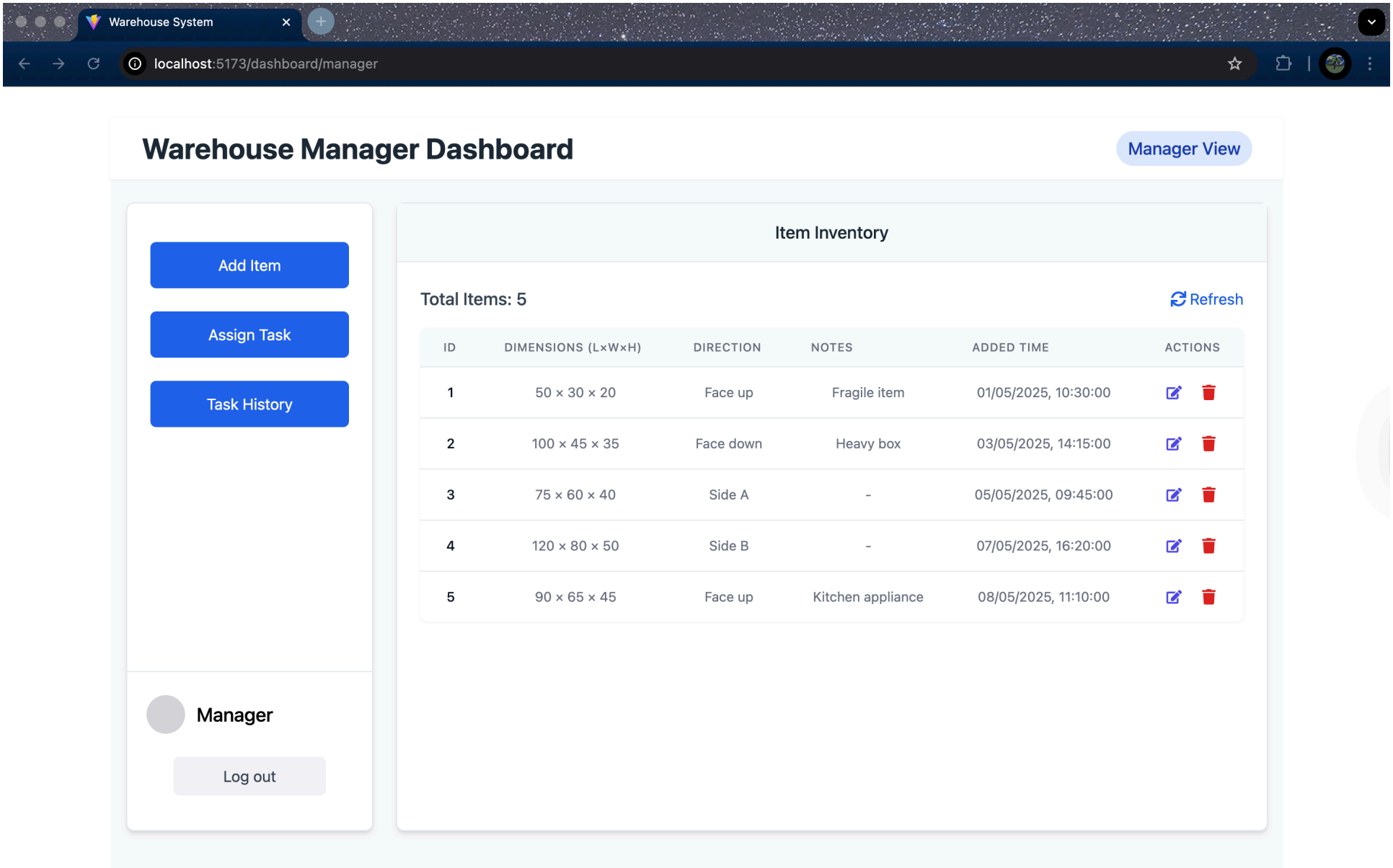

Figure 7 Manager Dashboard Page User Interface

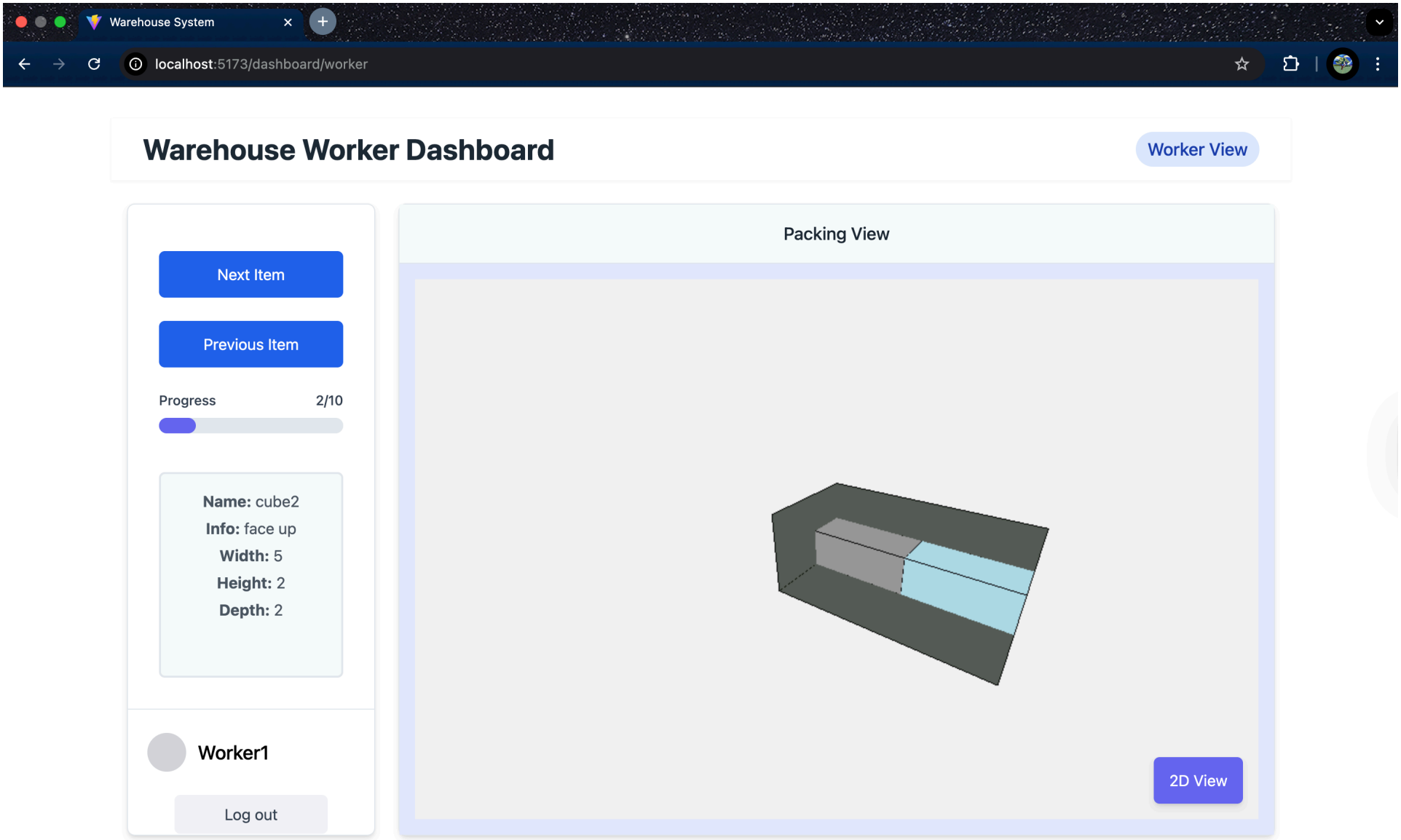

Figure 8 Manager Dashboard Page User Interface

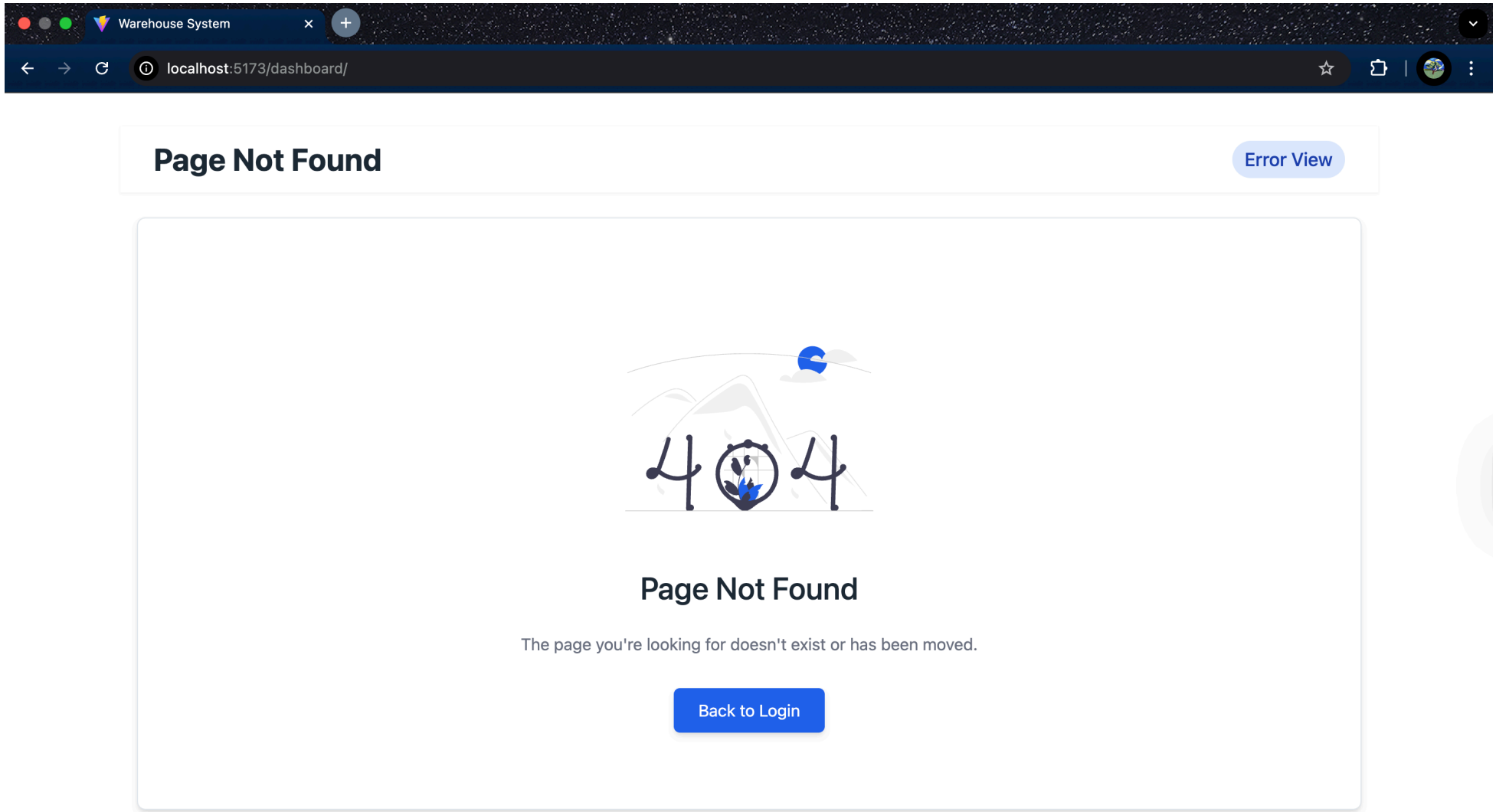

Figure 9 Not Found Page User Interface

Section 5 Project Backlog

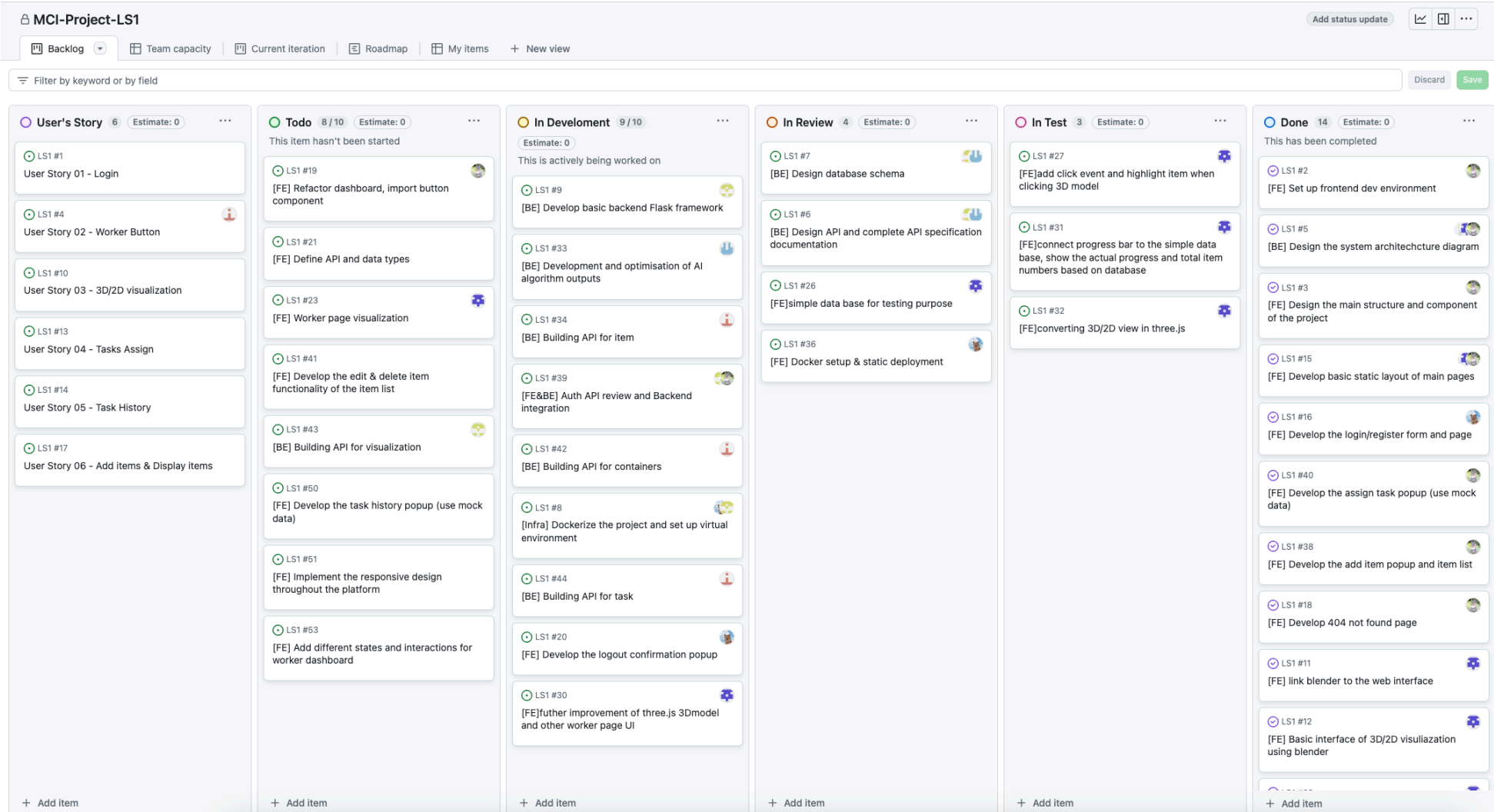

Figure 10 Progress Track and Project Management

Section 6 GitHub Pull Request

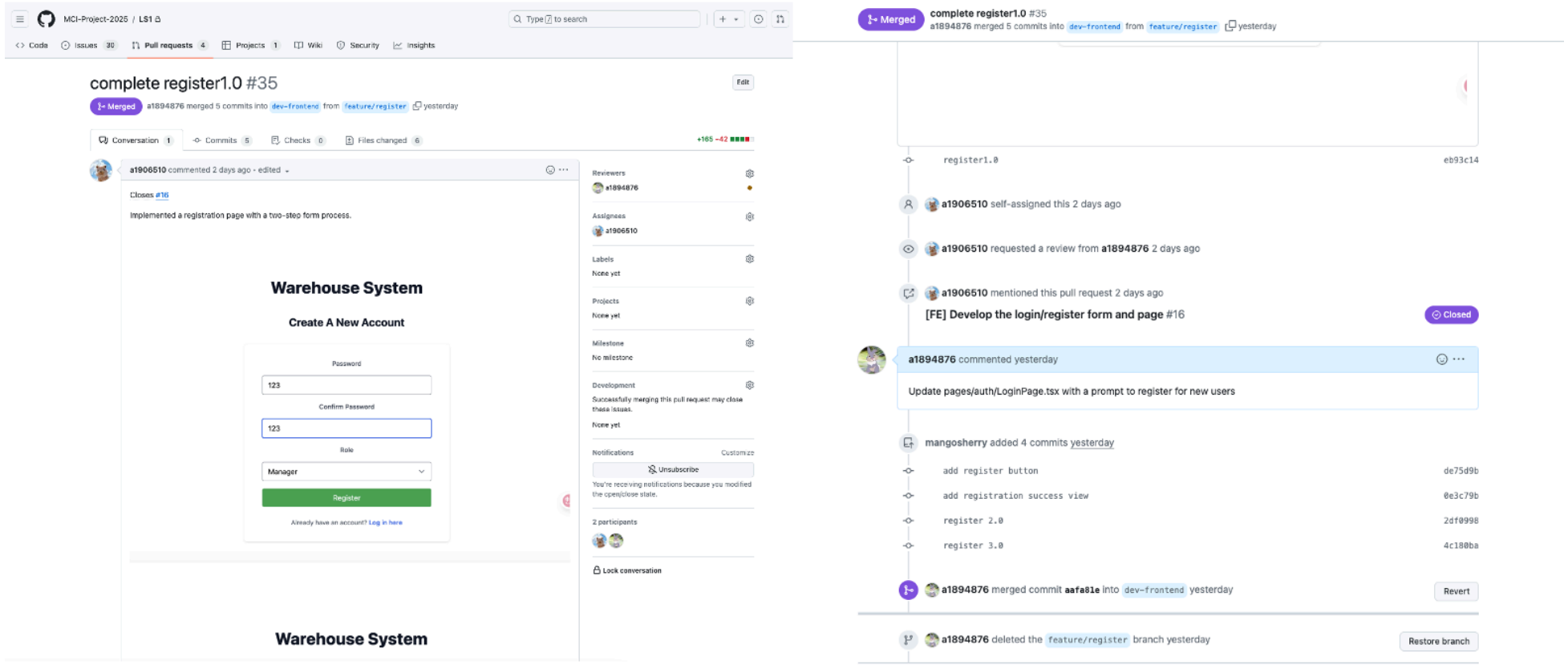

Figure 11 Version Control and Code Review (PR Example 1)

MCI-Project-2025 / LS1

Type / to search

Code | Issues 30 | Pull requests 4 | Projects 1 | Wiki | Security | Insights

Feature/item backend #45

Edit

Open a1909879 wants to merge 4 commits into `dev-backend` from `feature/item-backend`

Conversation 2 | Commits 4 | Checks 0 | Files changed 13 | +145 −0

a1909879 commented yesterday

Item Service

- Part of the Manager Console in the frontend.
- Handles API endpoints like:
  - POST /api/manager/add_item
  - PUT /api/manager/edit_item/{item_id}
  - DELETE /api/manager/delete_item/{item_id}
- Connects to the Item Repository in the database for full CRUD operations.
- Ensures each item is auto-named (Item0001, Item0002, etc.) and stored with dimensions, remarks, and orientation.

Calvin Chen added 4 commits yesterday

uploading the works from calvin	eaa61ee
uploading more works from calvin	a4fa913
Clean up removed API Learner_c folder and update backend code	63eb023
delete task.py from this branch	5909392

a1903270 commented yesterday

@a1909879

You added a bunch of backend files and folders directly under the root, which duplicates what's already in 'dev-backend'. It messes up the project structure and makes things harder to manage.

There's already a basic backend framework under '/backend'. Before adding new code, please make sure to understand the existing structure and build on top of it instead of creating a separate setup from scratch.

Please refactor your code to integrate it into the existing '/backend' base.

a1903270 commented yesterday

@a1909879 Since you've added new APIs, you should also include API documentation — explain how to use them, the naming conventions, and any expected request/response structure. You can add this doc directly to 'main' branch, such as: https://github.cs.adelaide.edu.au/MCI-Project-2025/LS1/blob/main/Technical_Docs/login_api_docs_v1.md

Reviewers
No reviews
Still in progress? Convert to draft

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.
None yet

Notifications Customize
Subscribe
You're not receiving notifications from this thread.

2 participants

Lock conversation

Figure 12 Version Control and Code Review (PR Example 1)