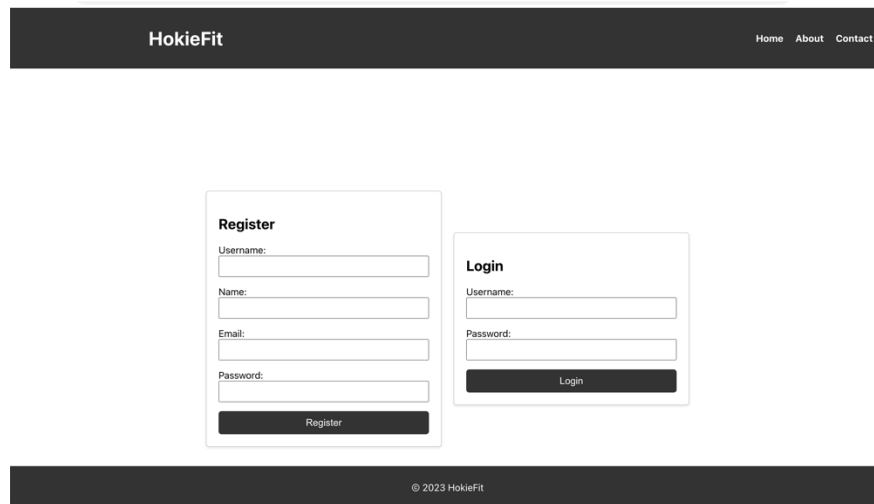


Sprint 1: HokieFit

1.0 Demo Working Product

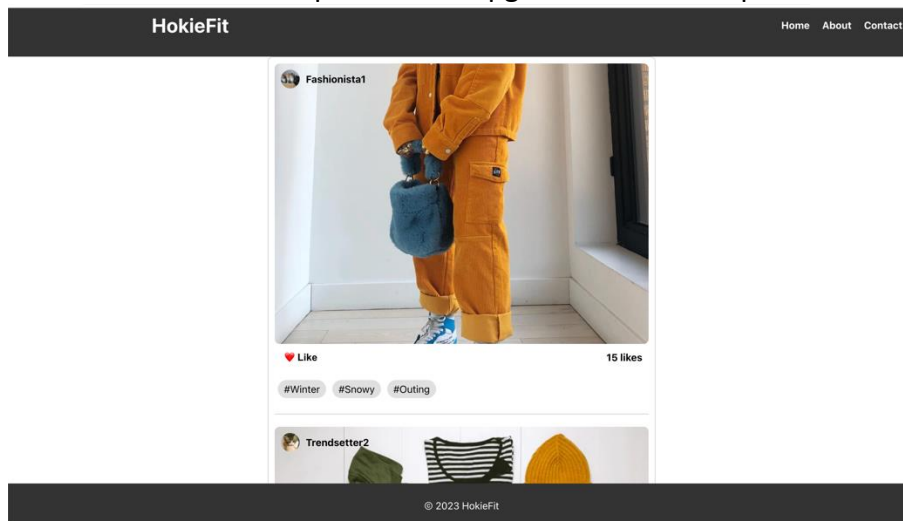
1. Register and Login

- The Registration and login pages are created in the same component. We have provided different components for headers and footers which can be used in all the other future components.



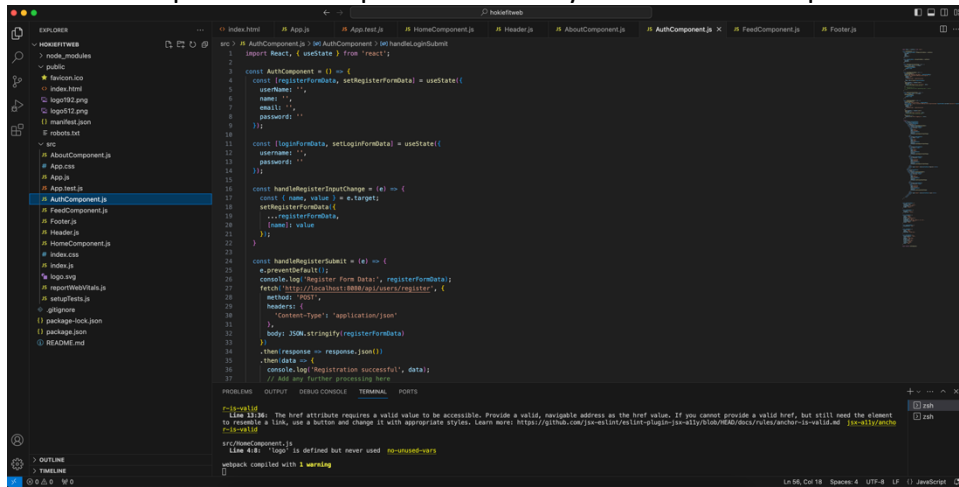
2. Homepage

- This is the home page which will list all the outfits, we will include search by tags and face detection options in this page in the future sprints.



3. Front end components

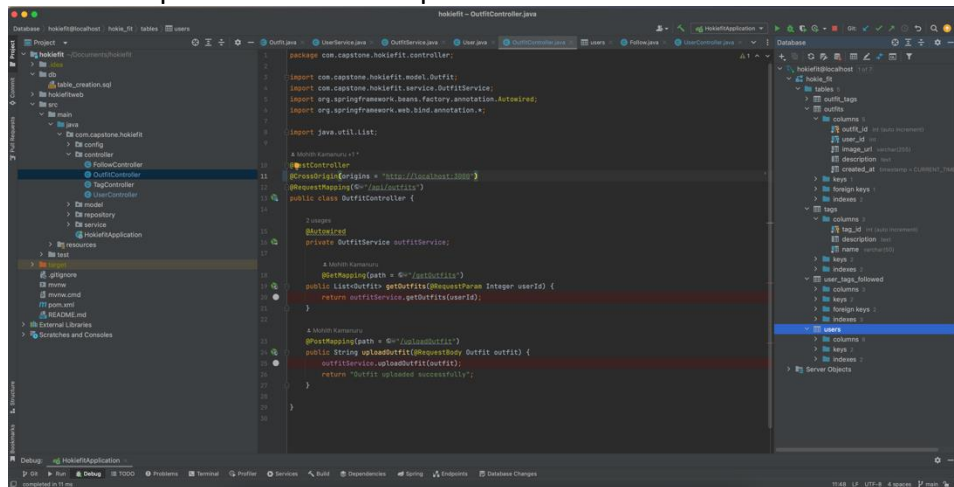
- Just an example of the component hierarchy for our frontend part of the application.



```
1 import React, { useState } from 'react';
2
3 const AuthComponent = () => {
4   const [loginFormData, setLoginFormData] = useState({
5     username: '',
6     password: ''
7   });
8
9   const [registerFormData, setRegisterFormData] = useState({
10    name: '',
11    email: '',
12    password: ''
13  });
14
15  const handleLoginSubmit = (e) => {
16    e.preventDefault();
17    console.log('Login Form Data:', loginFormData);
18    // Add API call for login logic here
19  };
20
21  const handleRegisterSubmit = (e) => {
22    e.preventDefault();
23    console.log('Register Form Data:', registerFormData);
24    // Add API call for registration logic here
25  };
26
27  return (
28    <div>
29      <h3>AuthComponent</h3>
30      <div>
31        <input type="text" value={loginFormData.username} />
32        <input type="password" value={loginFormData.password} />
33        <button type="button" value="Login" />
34      </div>
35      <div>
36        <input type="text" value={registerFormData.name} />
37        <input type="text" value={registerFormData.email} />
38        <input type="password" value={registerFormData.password} />
39        <button type="button" value="Register" />
40      </div>
41    </div>
42  );
43
44  </AuthComponent>
45
46  </div>
47
48  </div>
49
50  </div>
51
52  </div>
53
54  </div>
55
56  </div>
57
58  </div>
59
60  </div>
61
62  </div>
63
64  </div>
65
66  </div>
67
68  </div>
69
70  </div>
71
72  </div>
73
74  </div>
75
76  </div>
77
78  </div>
79
80  </div>
81
82  </div>
83
84  </div>
85
86  </div>
87
88  </div>
89
90  </div>
91
92  </div>
93
94  </div>
95
96  </div>
97
98  </div>
99
100 </div>
```

4. Backend and Database Tables

- List of potential tables (schemas) we might need in our database.
- The APIs implemented for this sprint

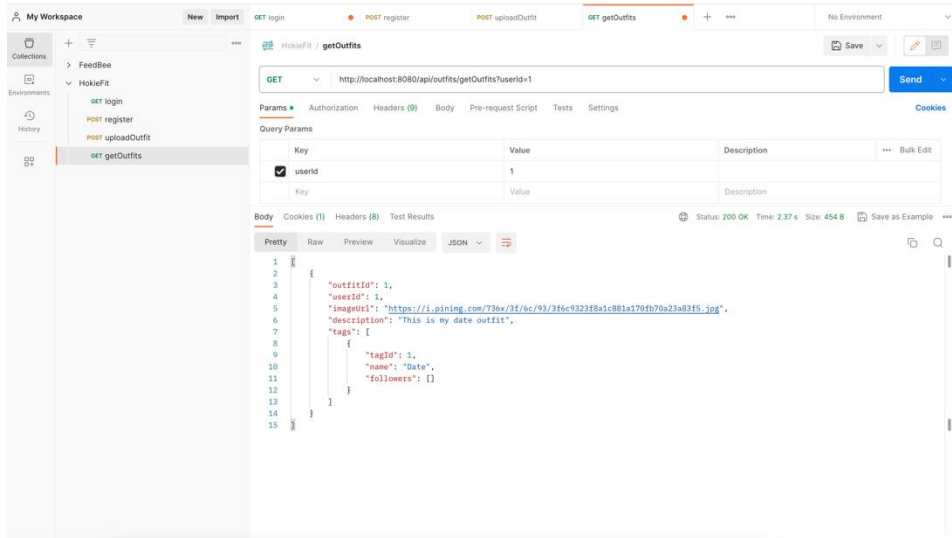


```
1 package com.capstone.hokierfit.controller;
2
3 import com.capstone.hokierfit.model.Outfit;
4 import com.capstone.hokierfit.service.OutfitService;
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.web.bind.annotation.*;
7 import java.util.List;
8
9 @RestController
10 @RequestMapping("http://localhost:8080")
11 @RequestMapping("api/outfits")
12 public class OutfitController {
13
14     @Autowired
15     private OutfitService outfitService;
16
17     @GetMapping("/{userId}")
18     public List<Outfit> getOutfits(@RequestParam Integer userId) {
19         return outfitService.getOutfits(userId);
20     }
21
22     @PostMapping("/{userId}")
23     public String uploadOutfit(@RequestBody Outfit outfit) {
24         outfitService.uploadOutfit(outfit);
25         return "Outfit uploaded successfully";
26     }
27
28 }
29
30 </OutfitController.java>
```

The database schema diagram shows the following tables and relationships:

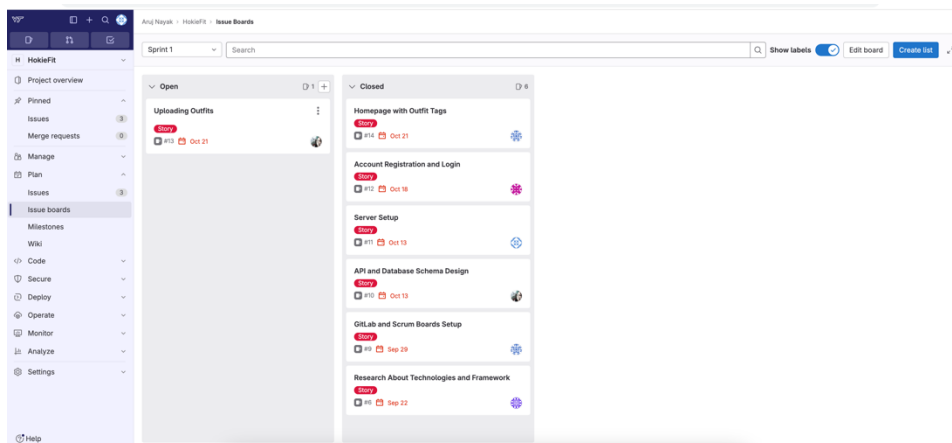
- users**: Contains columns for user_id, name, email, password, and created_at. It is the parent table for the Outfit table.
- outfits**: Contains columns for outfit_id, user_id, name, description, and created_at. It has a foreign key relationship with the users table.

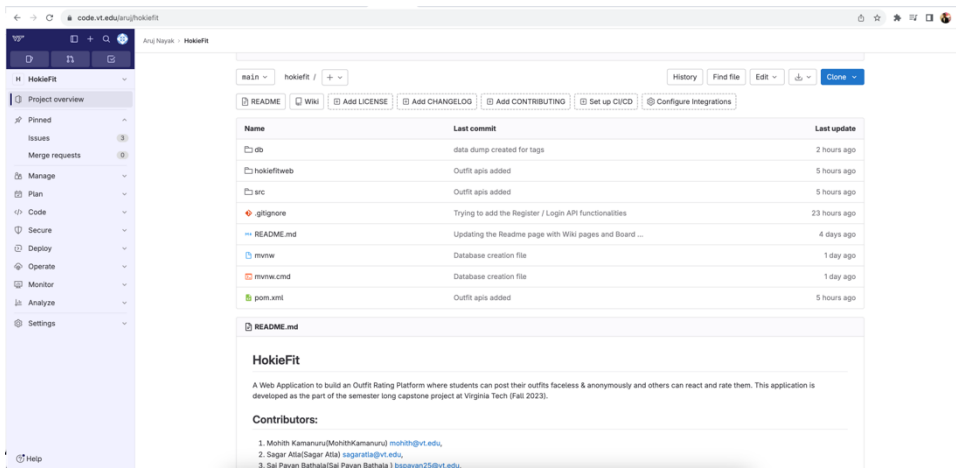
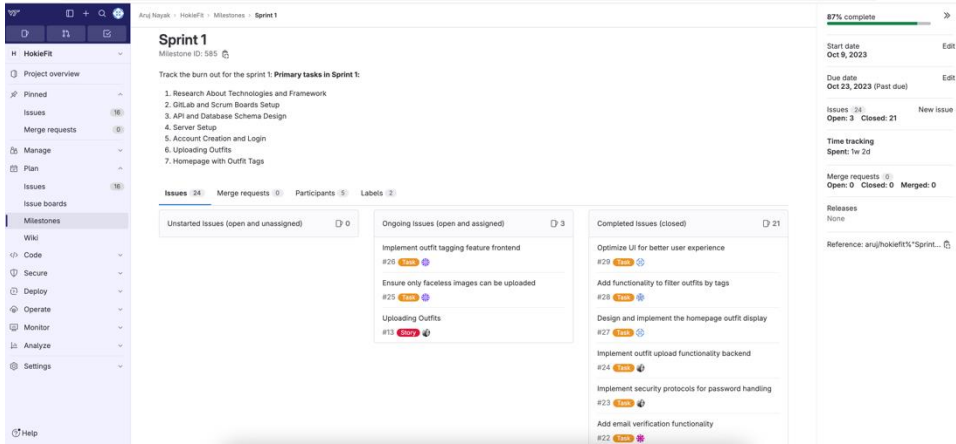
5. API testing using postman



2.0 Collaboration Space

1. Screenshots from scrum boards-





We are utilizing GitLab to streamline our development workflow effectively. It aids in issue tracking, prioritizing tasks, and managing sprints through milestones. Eventually, GitLab CI/CD will automate our testing and deployment, ensuring reliability and efficiency. Our GitLab project space, as showcased in the provided screenshot, is a hub of organized, transparent, and efficient collaboration, enabling us to track, manage, and execute tasks effectively, ensuring we are aligned with our project’s timelines and quality expectations.

3.0 Scrum

3.1 Sprint 1

1. Research About Technologies and Framework

- **Assignee:** Sai Pavan Bathala
- **Tasks:**
 - Evaluate potential frameworks and technologies - **Estimated Time:** 4 hours.

- Document the pros and cons of each technology - **Estimated Time:** 2 hours.

2. GitLab and Scrum Boards Setup

- **Assignee:** Aruj Nayak
- **Tasks:**
 - Set up the project repository on GitLab - **Estimated Time:** 1 hour.
 - Configure the scrum board and issue tracking - **Estimated Time:** 2 hours.

3. API and Database Schema Design

- **Assignee:** Sagar Atla
- **Tasks:**
 - Design the API endpoints structure - **Estimated Time:** 3 hours.
 - Create the database schema - **Estimated Time:** 4 hours.

4. Server Setup

- **Assignee:** Mohith Kamanuru
- **Tasks:**
 - Set up the development server - **Estimated Time:** 2 hours.
 - Configure the environment for Spring Boot and React - **Estimated Time:** 3 hours.

5. Account Creation and Login

- **Assignee:** Srikanth Karri
- **Tasks:**
 - Implement user registration using VT credentials - **Estimated Time:** 4 hours.
 - Add email verification functionality - **Estimated Time:** 3 hours.
 - Implement security protocols for password handling - **Estimated Time:** 2 hours.

6. Uploading Outfits

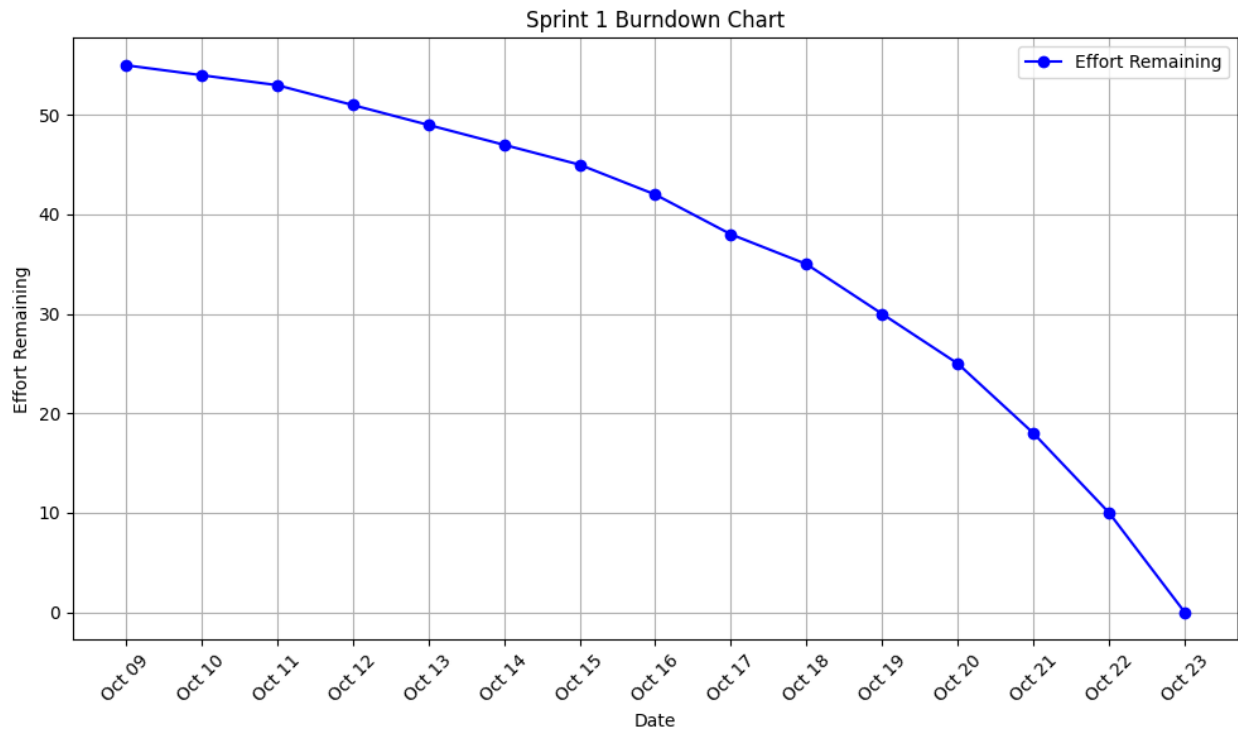
- **Assignee:** Sagar Atla
- **Tasks:**
 - Implement outfit upload functionality backend - **Estimated Time:** 5 hours.
 - Ensure only faceless images can be uploaded - **Estimated Time:** 3 hours (Not Completed).
 - Implement outfit tagging feature frontend - **Estimated Time:** 4 hours (Not Completed).

7. Homepage with Outfit Tags

- **Assignee:** Aruj Nayak
- **Tasks:**
 - Design and implement the homepage outfit display - **Estimated Time:** 6 hours.

- Add functionality to filter outfits by tags - **Estimated Time:** 4 hours.
- Optimize UI for better user experience - **Estimated Time:** 3 hours.

3.2 Burndown Chart Please also include the ideal burndown. This can be just something like



3.3 Sprint Retrospective

What was accomplished:

1. We were successful in setting up the development environment, including the server, GitLab, and Scrum boards.
2. The user registration and login, email verification, and security protocols for password handling were also completed.
3. The homepage design was completed, but filtering by tags is still in progress.

What was not accomplished:

We couldn't complete the outfit upload functionality. The backend was developed but integrating it with the frontend remains.

Impediments and Solutions:

1. We faced challenges in implementing the faceless image upload, which led to a delay. We plan to allocate more time and resources to this in the next sprint.
2. The frontend development was slower than expected. We will be looking into additional tools and libraries to speed up the process.

What went well:

1. Collaboration and communication among the team were effective.
2. The server and development environment setup were smooth.

What could be improved:

1. Task estimation needs to be more accurate to avoid spillovers.
2. We need to improve our frontend development speed.

3.4 Product Backlog(updated)

During the sprint retrospective, we took the opportunity to review and update our product backlog. We reassessed the initial user stories and tasks to align with the challenges and learnings encountered during the first sprint. Below are the newly identified user stories, along with any that have been deprioritized or removed.

Newly Identified User Stories:

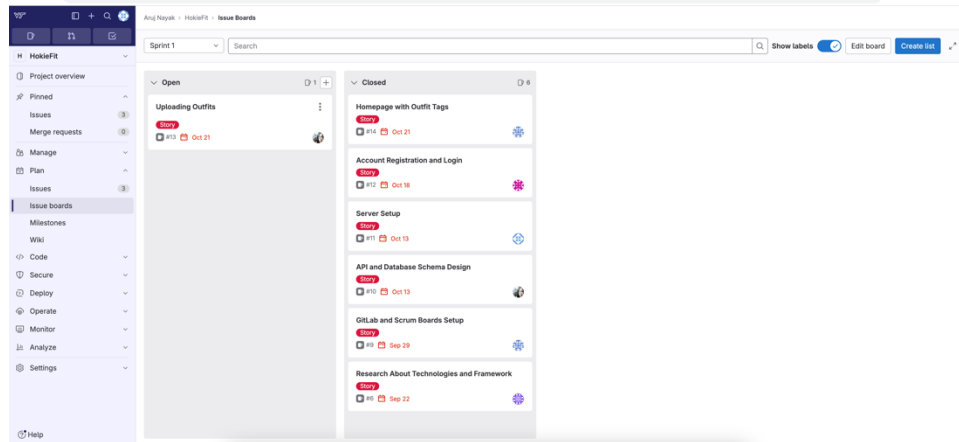
1. Enhanced Image Upload:

- **Description:** To make the outfit upload process more user-friendly and versatile, we plan to enhance the feature to support multiple image formats and sizes. This improvement aims to enhance user engagement and content diversity on the platform.
- **Tasks:**
 - Implement support for various image formats.
 - Optimize image loading speed for a better user experience.

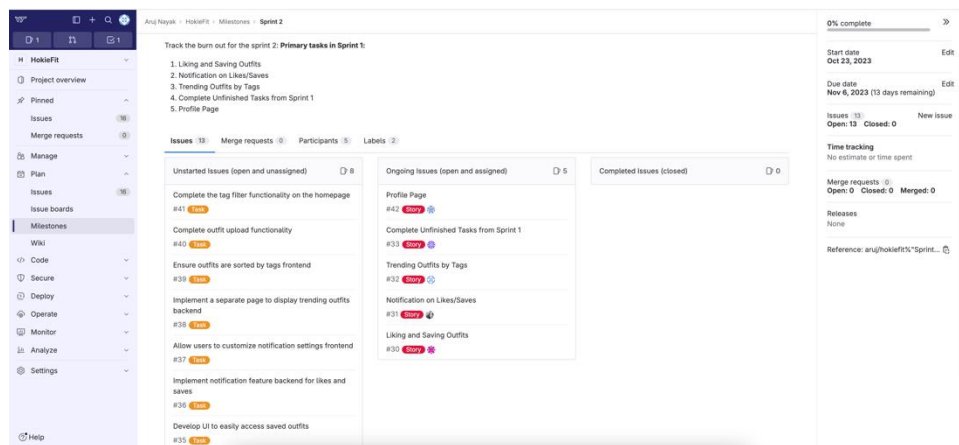
2. User Profile Customization:

- **Description:** We identified the need for users to have the ability to personalize their profiles. This feature will include options to choose themes, display names, and other customizable elements to make the user experience more engaging.
- **Tasks:**
 - Implement theme selection options.
 - Add features for customizing display names and profile appearances.

3.5 Issue tracking



3.6 Sprint 2 Backlog



1. Liking and Saving Outfits

- **Assignee:** Srikanth
- **Tasks:**
 - Implement like and save feature for outfits backend - **Estimated Time:** 4 hours.
 - Develop UI to easily access saved outfits - **Estimated Time:** 3 hours.

2. Notification on Likes/Saves

- **Assignee:** Sagar
- **Tasks:**
 - Implement notification feature backend for likes and saves - **Estimated Time:** 5 hours.
 - Allow users to customize notification settings frontend - **Estimated Time:** 3 hours.

3. Trending Outfits by Tags

- **Assignee:** Mohith
- **Tasks:**
 - Implement a separate page to display trending outfits backend - **Estimated Time:** 4 hours.
 - Ensure outfits are sorted by tags frontend - **Estimated Time:** 3 hours.

4. Complete Unfinished Tasks from Sprint 1

- **Assignee:** Sai Pavan, Aruj
- **Tasks:**
 - Complete outfit upload functionality - **Estimated Time:** 4 hours.
 - Complete the tag filter functionality on the homepage - **Estimated Time:** 2 hours.