**IFN636: Software Life Cycle Management:**

**Assessment 1**

Total Marks 20

**Assessment name:** Software requirements analysis and design (Full-Stack CRUD Application Development with DevOps Practices)

**We have provided this template for you. Include all parts of this assessment in one file (this file), convert it to PDF, and then submit it via Canvas before the deadline.**

**The first page (this page: cover page) of your assessment file should include the following information:**

**Mark Distribution:**

**Marks**

* Project design with SysML 3
* Project Management with JIRA 4
* Backend Development (Node.js + Express + MongoDB) 3
* Frontend Development (React.js) 2
* GitHub Version Control & Branching Strategy 2
* CI/CD Pipeline Setup 5
* README.md and Report 1

Total Marks: 20

**Full name: \_\_\_\_\_\_\_\_\_\_\_** **Jenish Patel \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Student ID: \_\_\_\_\_\_\_\_\_** **n12124371 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Tutor’s name (tutor’s full name): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Tutorial day and time: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

EC2 Instance Name and ID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

EC2 Instance Link: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Project Overview

Online Recruitment and Assessment System (HireHub) is an all-inclusive hiring tool that is aimed at simplifying the task of recruiting employees. It allows employers to advertise vacancies, screen the received applications by means of automated screening tests, organise interviews, and keep track of the candidates' progress. The development of the system relied on the application of a full software development lifecycle, applying backend and frontend in a modular and scalable frame. The backend, which was developed using Node.js, Express, and MongoDB, performs authentication, job control, applications, assessment and interview scheduling. Frontend created in React.js gives an intuitive interface where employers and applicants can have role-based access. The SysML diagrams fully embodied functional and non-functional requirements, whereas JIRA took care of sprints, backlogs, and user stories. The automatic building, testing, and deployment are achieved with GitHub version control and CI/CD pipelines using GitHub actions on AWS EC2, and there is no conflict. The end product is a full-stack cloud-deployed platform developed and built upon professional software engineering practices; it is well tested and has the potential to be used in actual recruitment.

# Project Design

## 2.1 Requirement Diagram using SysML

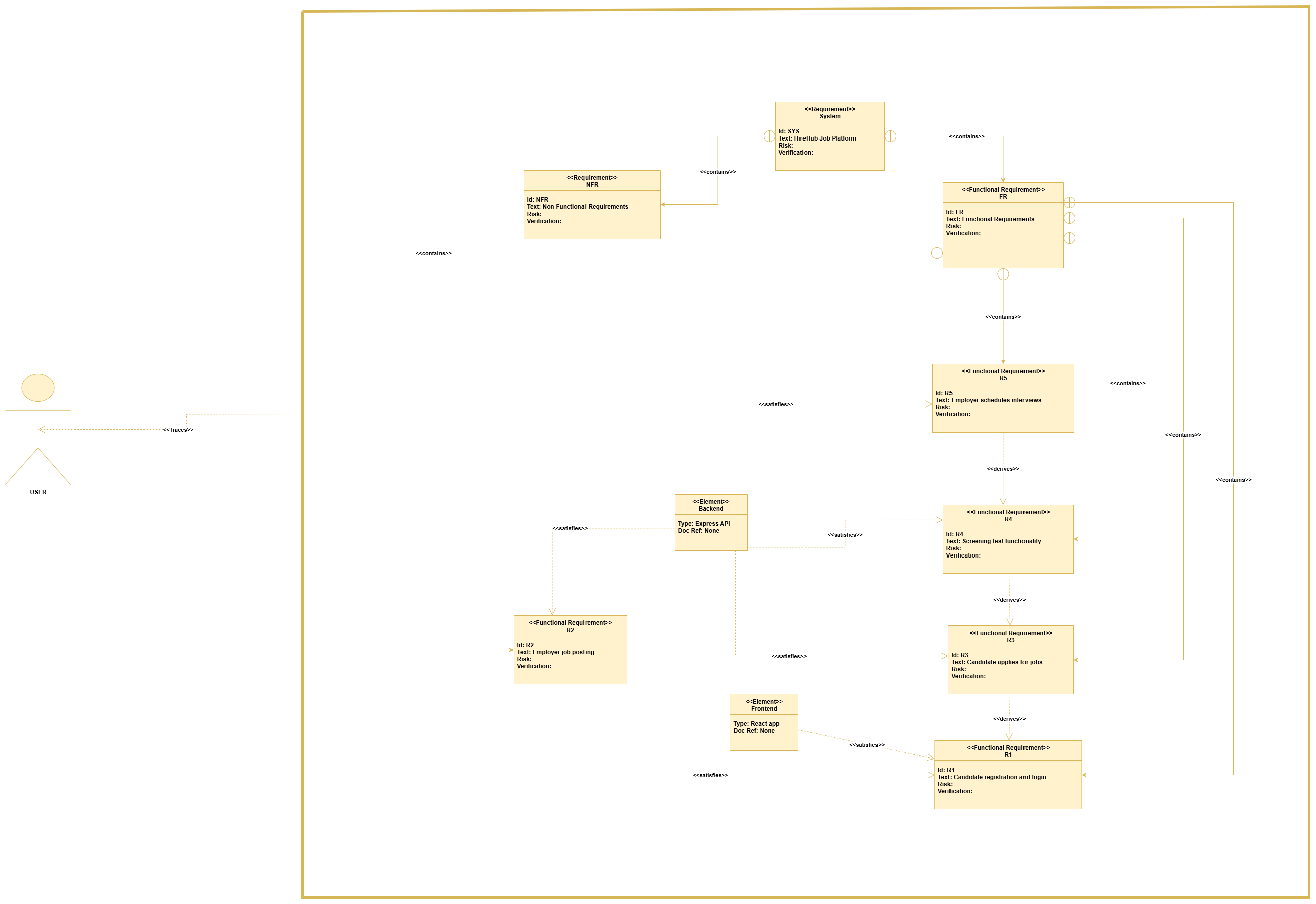


Figure : Requirement diagram

The Requirement Specification Diagram of the HireHub Job Platform presents the high-level requirements of the system and their relations. As shown in the diagram, the HireHub Job Platform is the base system, and it is subdivided into two categories: Functional Requirements (FR) and Non-Functional Requirements (NFR).

The functional requirements explain the main characteristics of the platform, namely: registration and log in to the platform by the candidates, posting of jobs on the platform by the employers, application of jobs by the candidates on the platform, the screening test feature on the platform, and the posting of a meeting with the candidates by the employers on the platform (R5). These requirements have derived relationships depicting their interdependency; an example, candidates will not be able to apply to jobs until they have registered and logged in. In the same way, interviews cannot take place without the completion of the screening tests.

System components used in the achievement of these requirements are also integrated into the diagram. The Frontend (React app) can utilise user-facing features like registration and job application, and the Backend (Express API) can deal with job postings, processing of a job application, screening processes, and schedule interviews.

With this formal visualisation, there is a clear connection between requirements and their achievement to plan better, trace, and validate the whole project. It is useful in making stakeholders aware of what the system must do as well as how every requirement is satisfied in the architecture of the system (VisualParadigm, 2023).

## 2.2 Block Definition Diagram using SysML

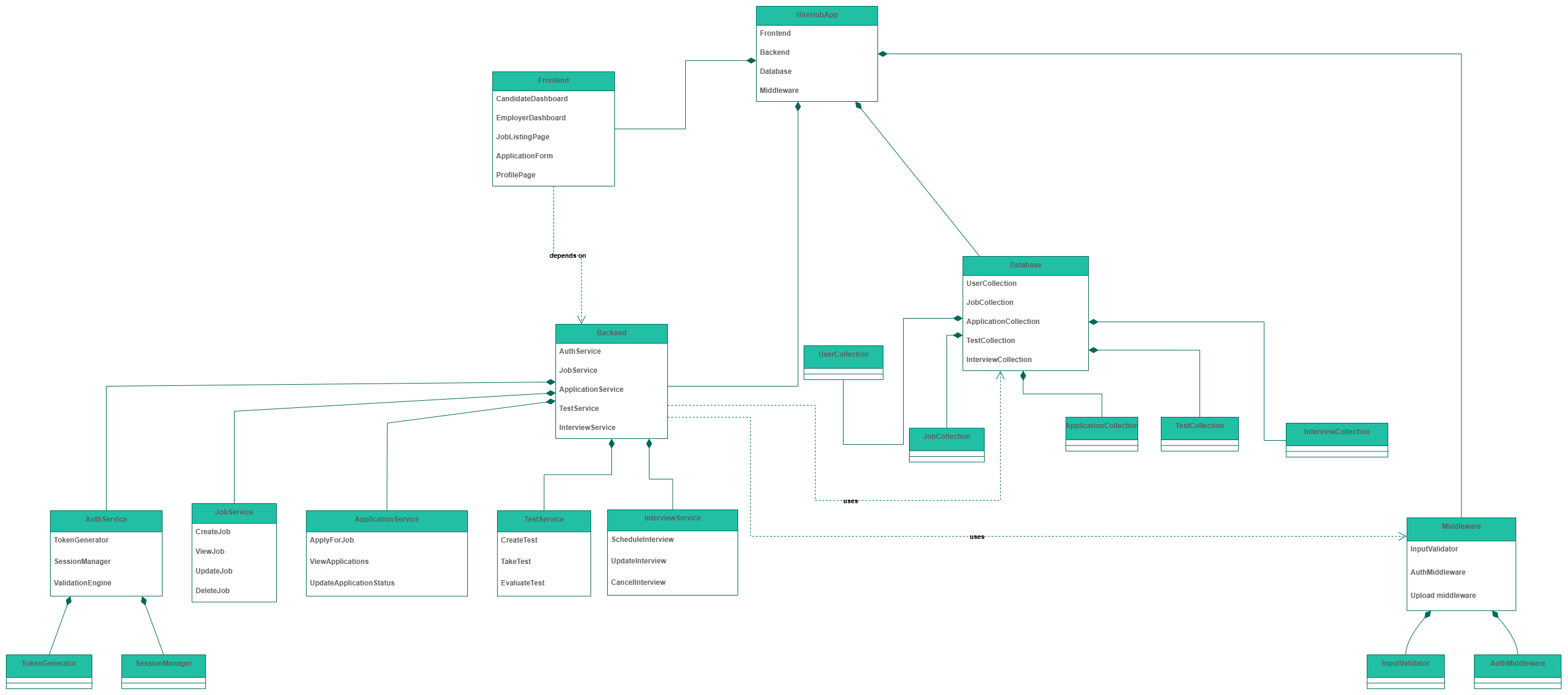


Figure Block Definition diagram

The Block Definition Diagram (BDD) on the HireHub Job Platform visualises the structure of the system and the relationship between the main components of the system. The highest level consists of the HireHubApp, which is constituted of the Frontend, Backend, Database and Middleware.

The Frontend block consists of all viewable elements such as the Candidate Dashboard, the Employer Dashboard, the Job Listing Page and the Application Form, as well as the Profile Page. These offer the primary user interface between the candidates and employers regarding the platform.

The Backend block implements the core business logic of the application and is subdivided into a number of services: AuthService (authentication, generation of tokens, and sessions management), JobService (job creation, updates, and deletions), ApplicationService (job applications and status changes), TestService (provisioning and marking screening tests) and InterviewService (interview scheduling and updates) (Walker, 2025).

The Database block is a persistent collection of data, having several collections: Users, Jobs, Applications, Tests, and Interviews have consistent data and data retrieval.

In the Middleware block, some utility elements such as InputValidator and AuthMiddleware guarantee validation of requests and secure access throughout the system.

As shown in the dependency relations, the Frontend is dependent on the Backend, and the Backend depends on the Database, but uses Middleware to implement validation and authentication routines.

## 2.3 Parametric Diagram using SysML

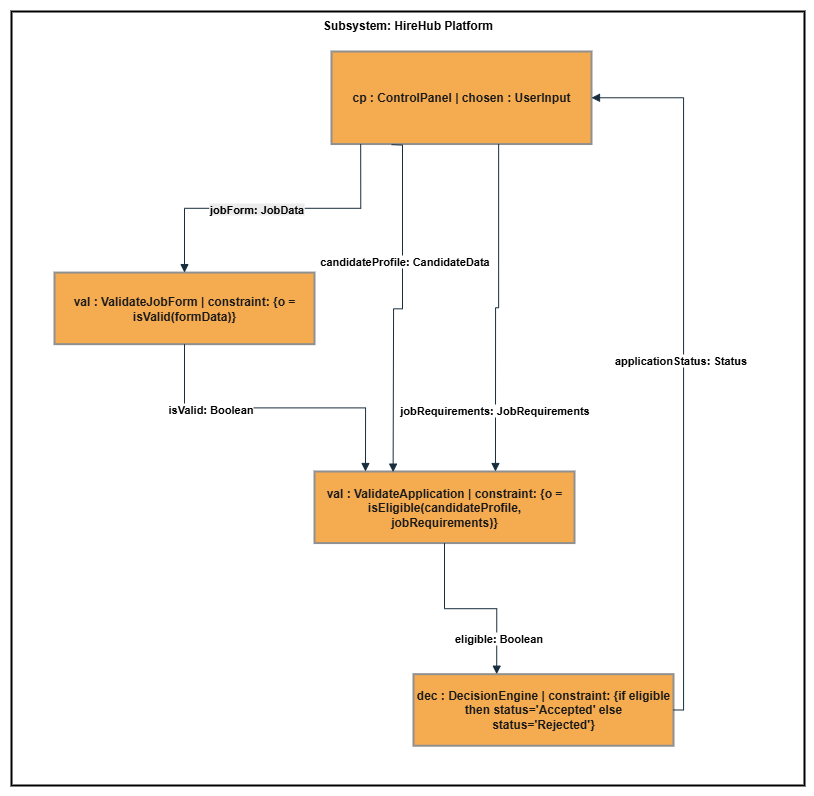


Figure Parametric Diagram

The Parametric Diagram of the HireHub platform represents the logical flow and limitations of processing the job offer of a candidate. The diagram is specifically directed towards data flows between components and the imposition of constraints upon them in order to achieve application results.

In essence, the ControlPanel (CP) block is the main interface by which the input of the user is gathered, including the job form, candidate profile, and job requirements. This information is relayed to constraint blocks such where they are validated, and decisions are made.

ValidateJobForm (VAL1) constraint provides that the job form details given are within specifications. It returns a Boolean value (isValid) that is an indication of the form being filled properly.

Once passed, the information is diverted to ValidateApplication (VAL2), which checks whether the candidate profile fits the requirements of the job. The result of this is a valid Boolean value (Bruckner, 2023).

The DecisionEngine (DEC) constraint makes the final decision and will examine the valid value and give the application an appropriate status, whether it is “Accepted” or “Rejected.” The result is then returned to the Control Panel to be further processed or presented to the user.

This type of parametric model takes into account the effects of dependency of operations as well as the limitations that guide the decision-making process, and, in jargon, it ensures a consistent, rules-based assessment of all applications in the HireHub system.

# Project Management (JIRA)

## 3.1 Provide a public link to the Jira Board or project link

<https://jenishfootball.atlassian.net/jira/software/projects/ORASH/boards/35?atlOrigin=eyJpIjoiYTIzMTFhMTFkNGQ4NDI1OGEyMjJkNDMyZTM0NzM2ZWIiLCJwIjoiaiJ9>

## 3.2 Product Backlog Screenshot

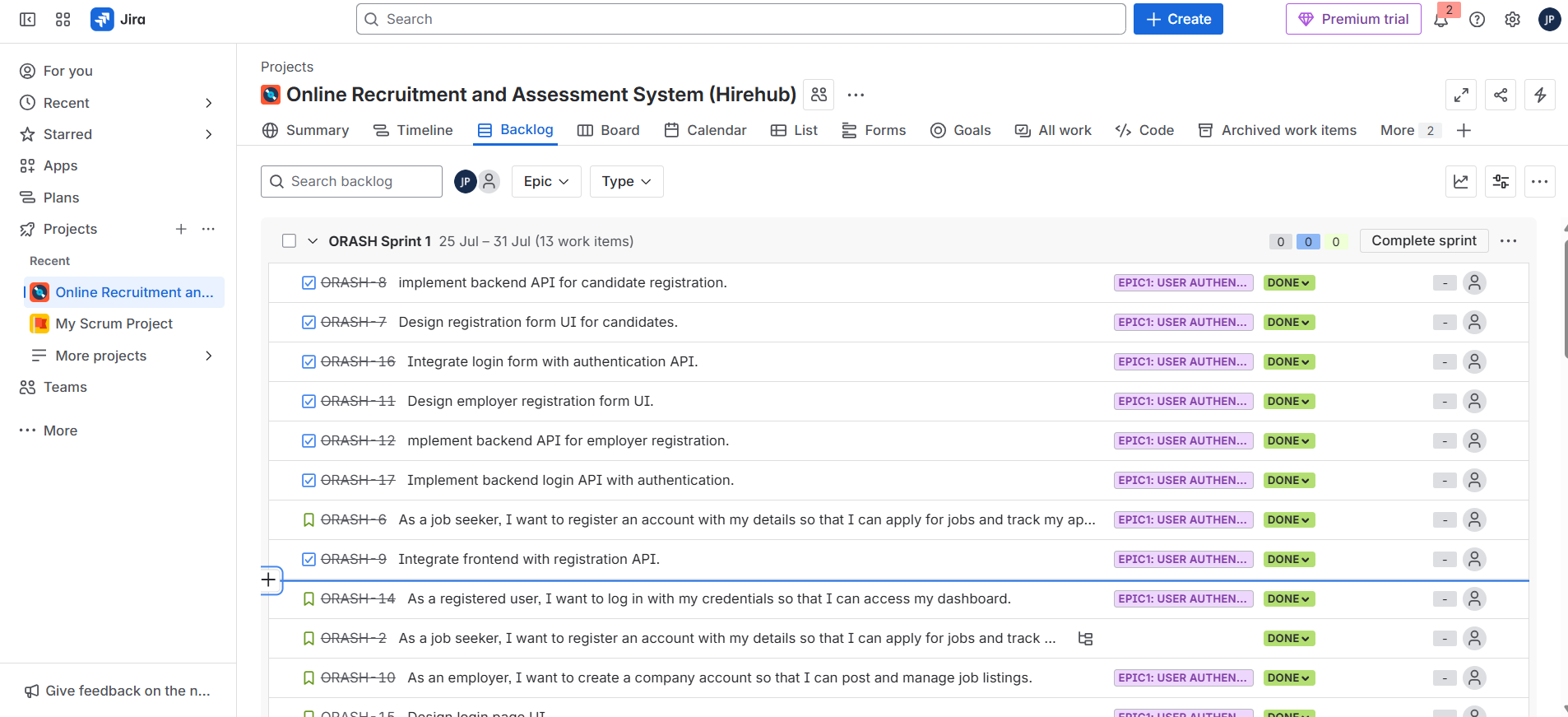


Figure : Sprint 1 Backlog

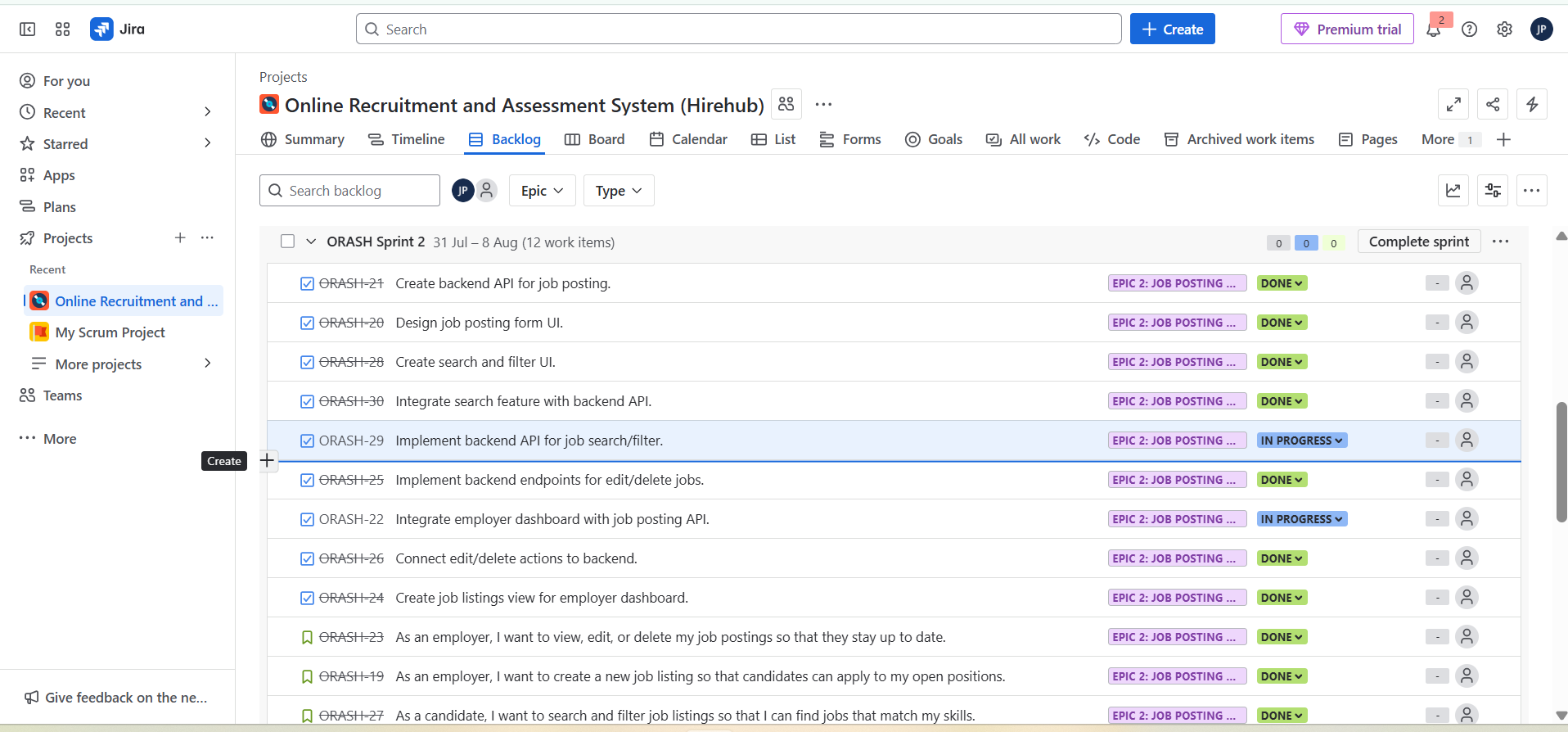


Figure : Sprint 2 Backlog

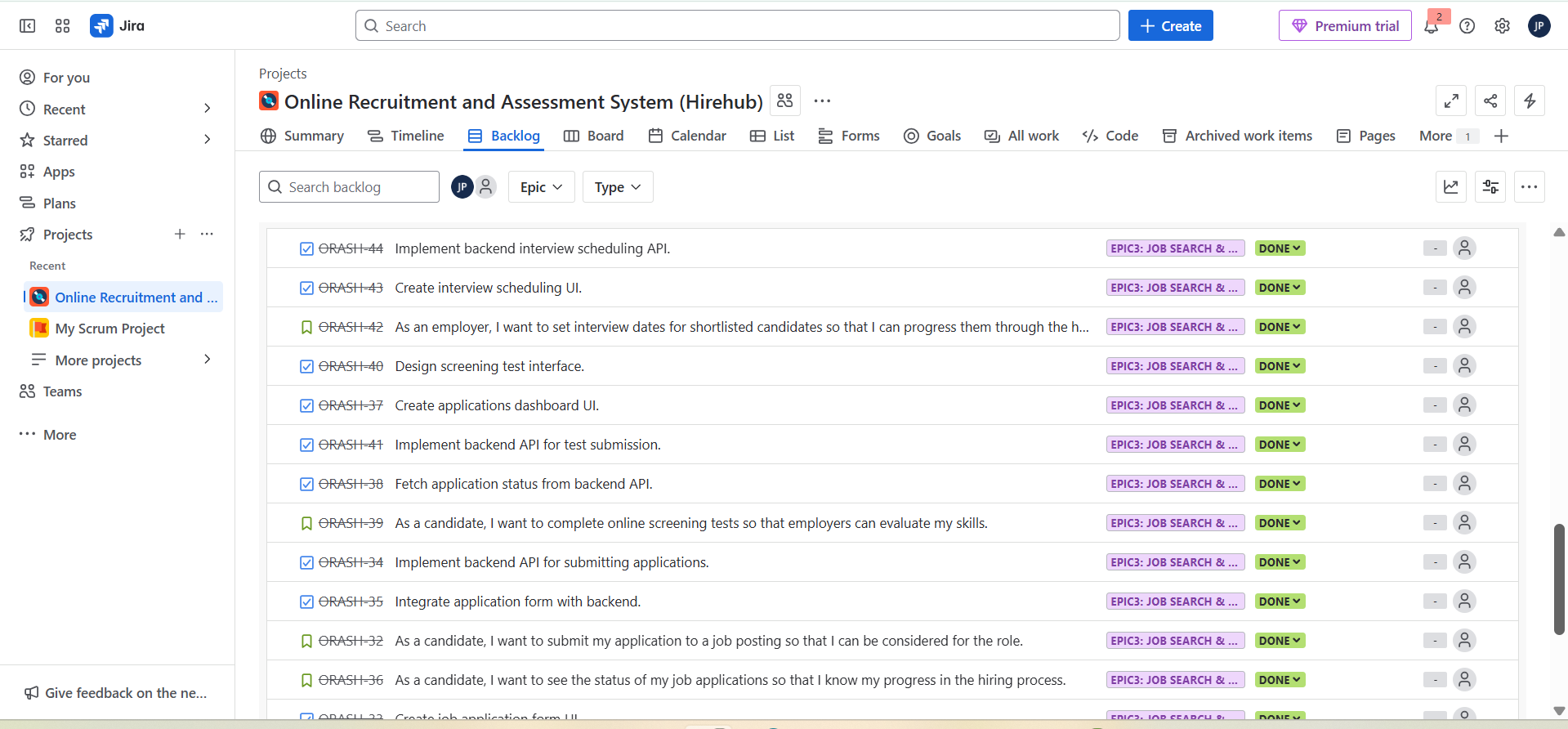


Figure : Sprint 3 backlog

## 3.3 Project Timeline Screenshot (Include Epic, User Story)

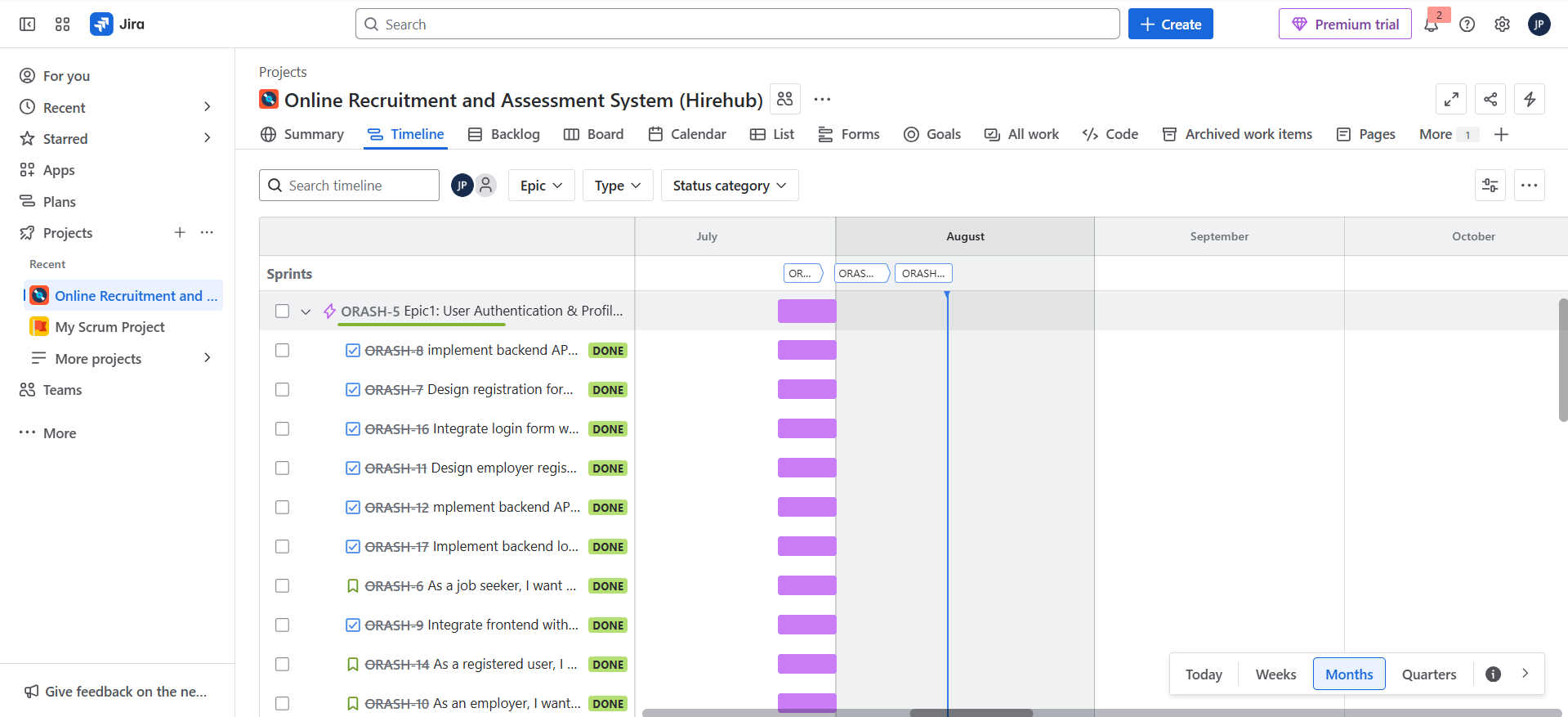


Figure Project Timeline Screenshot 1

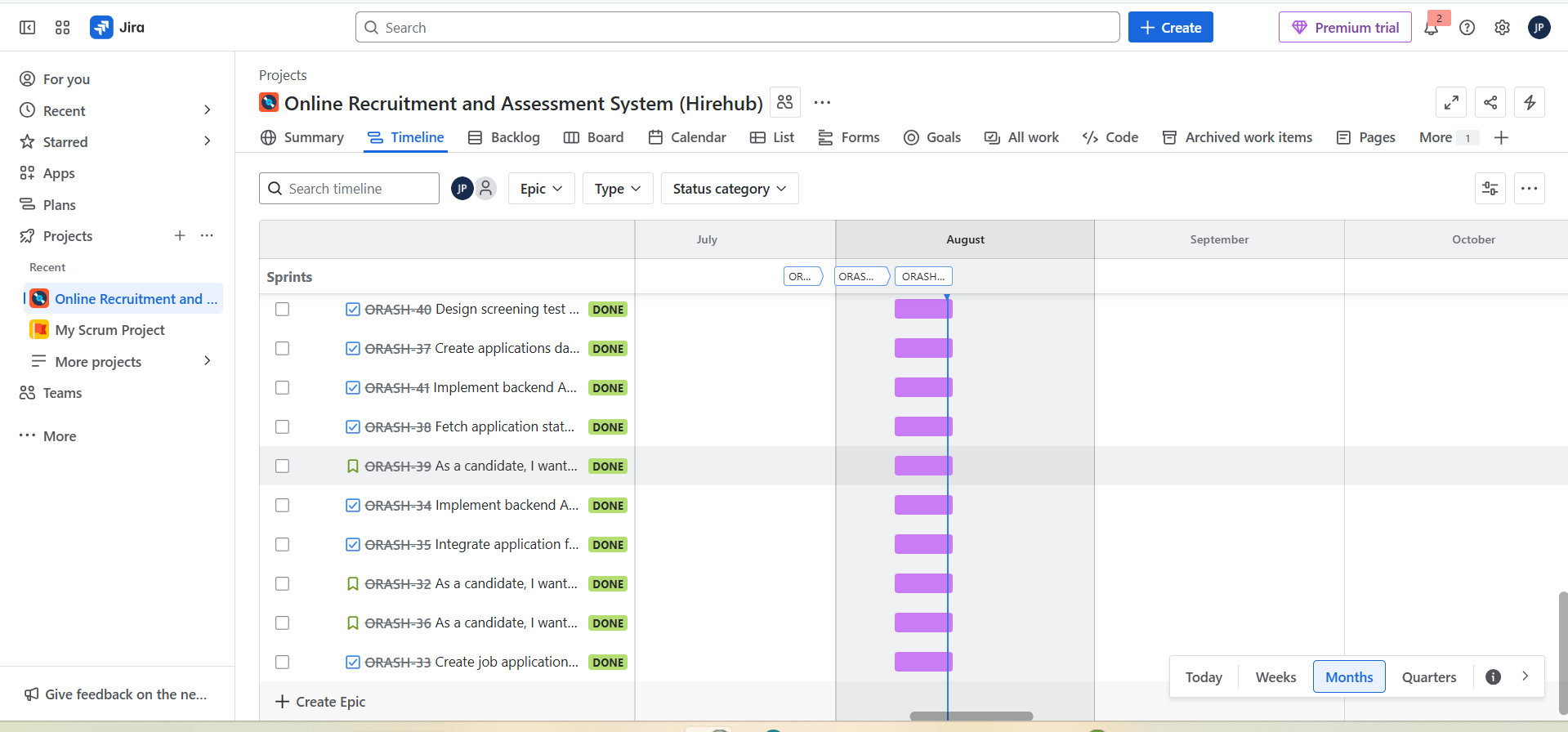


Figure Project timeline screenshot 2

## 3.4 Provide one Screenshot of any one User Story, including subtasks

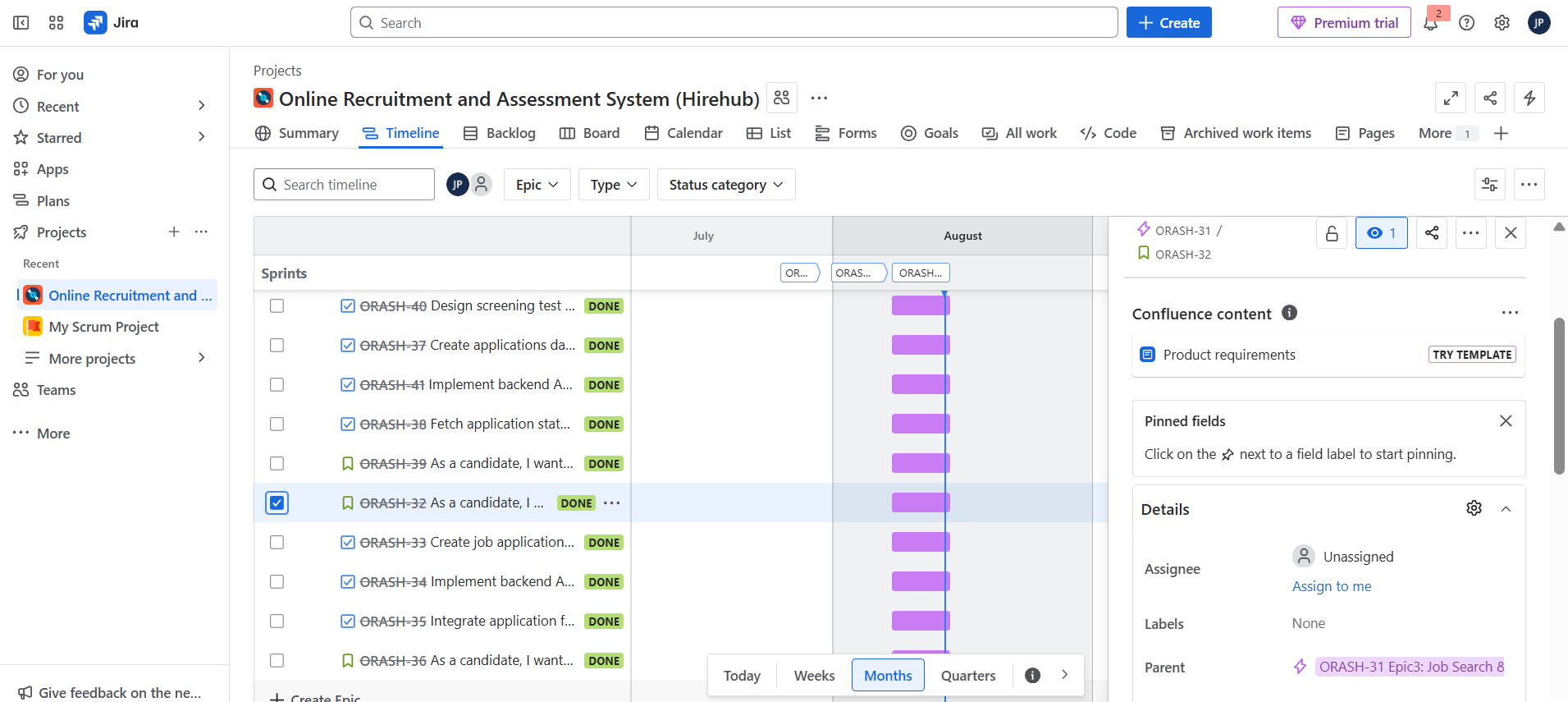


Figure Screenshot shows ORASH-32 is a user story, and 33,34,35 their tasks.

## 3.5 Provide a Screenshot of where you planned all sprints

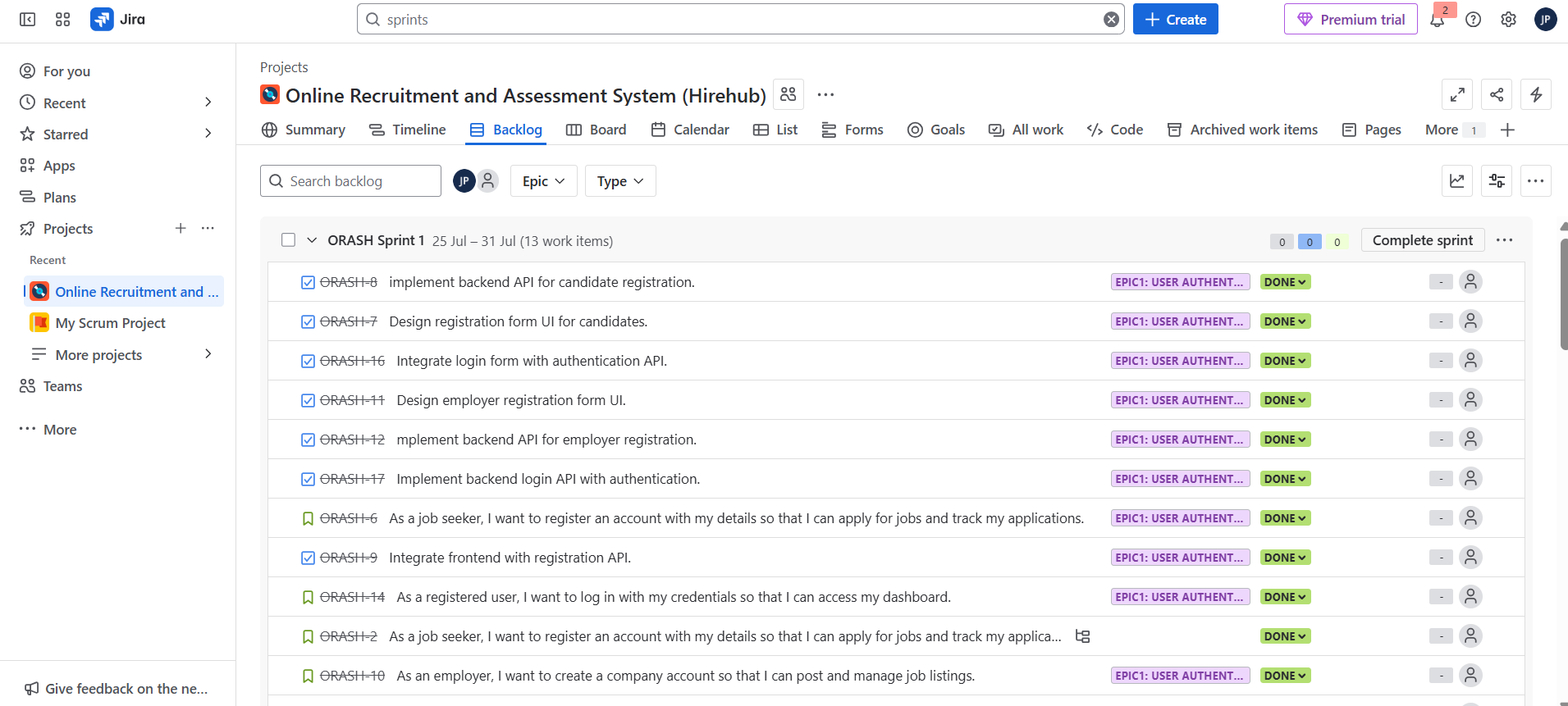


Figure : Planned Sprint 1

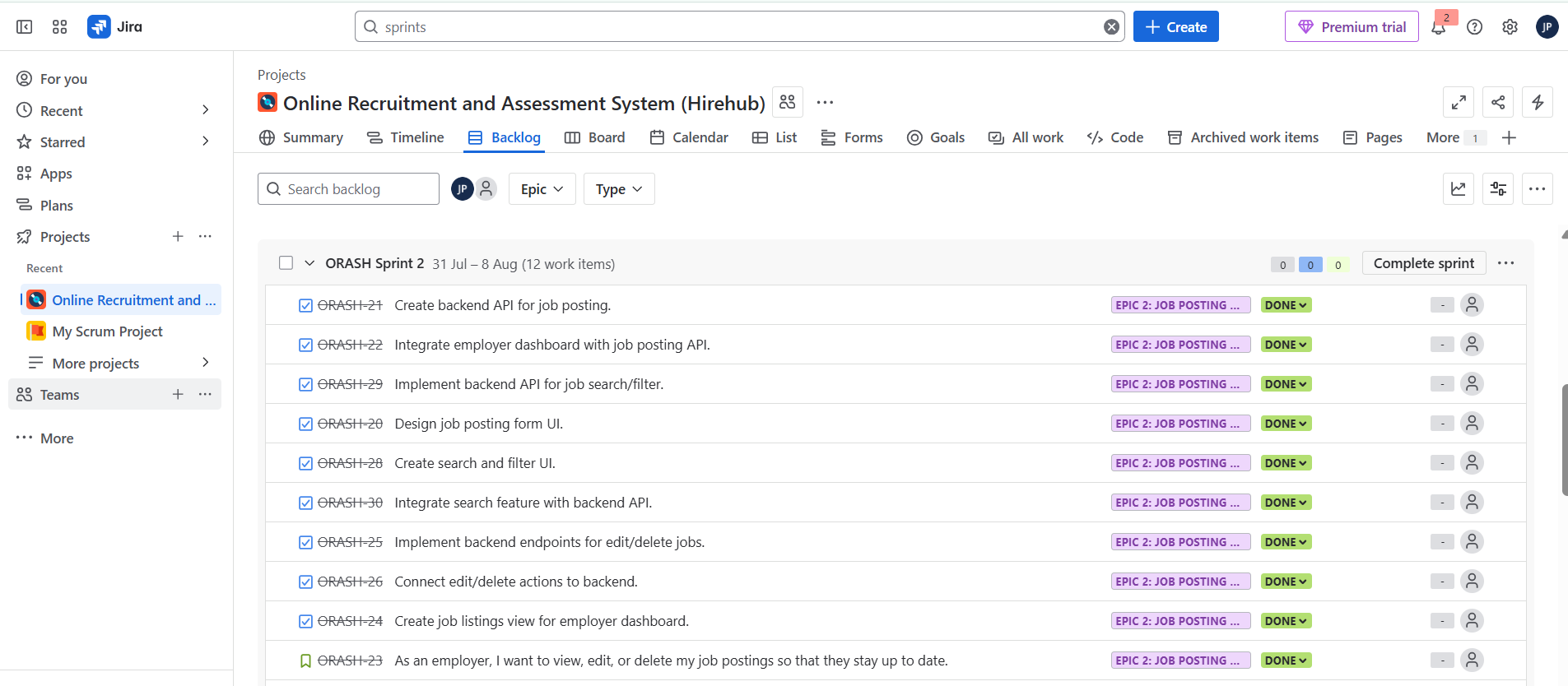


Figure Planned Sprint 2

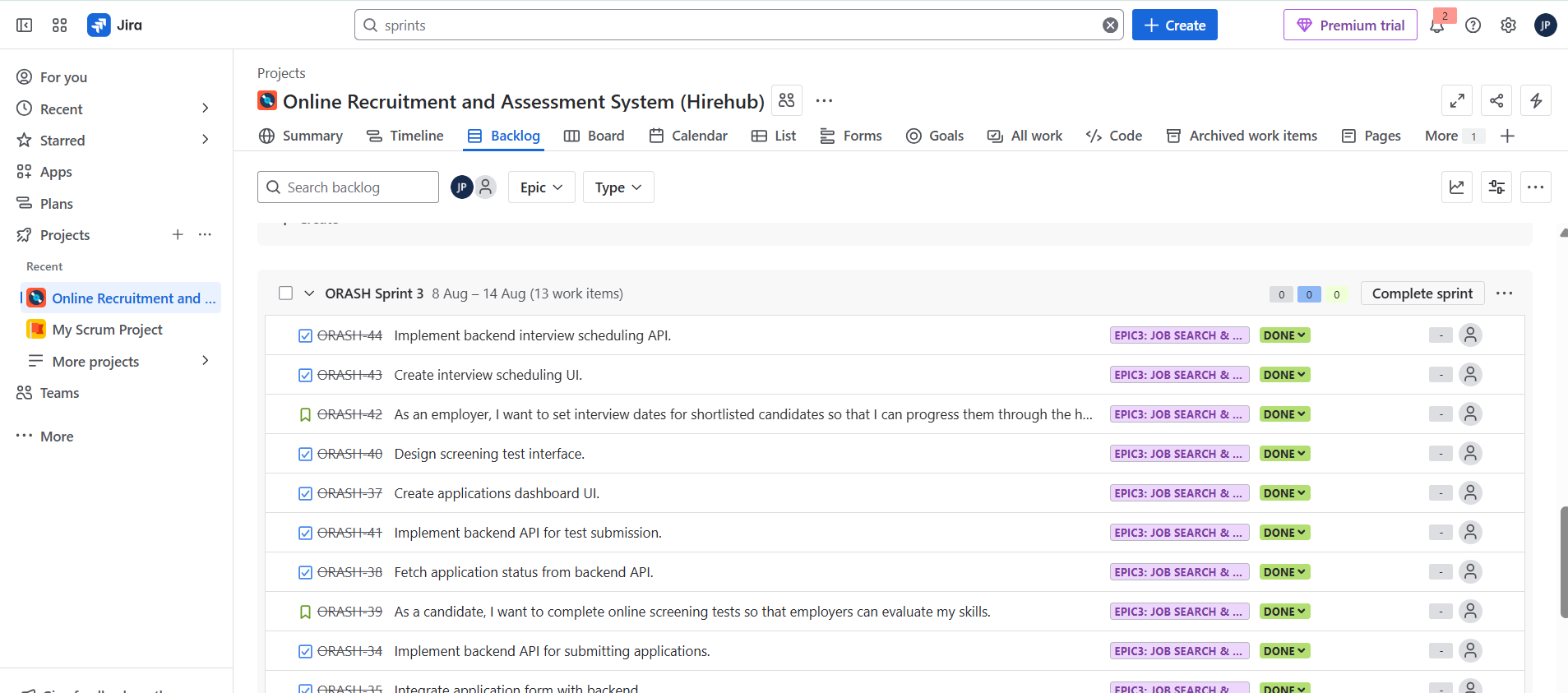


Figure Planned Sprint 3

## 3.6 Provide a Screenshot of where you started one sprint

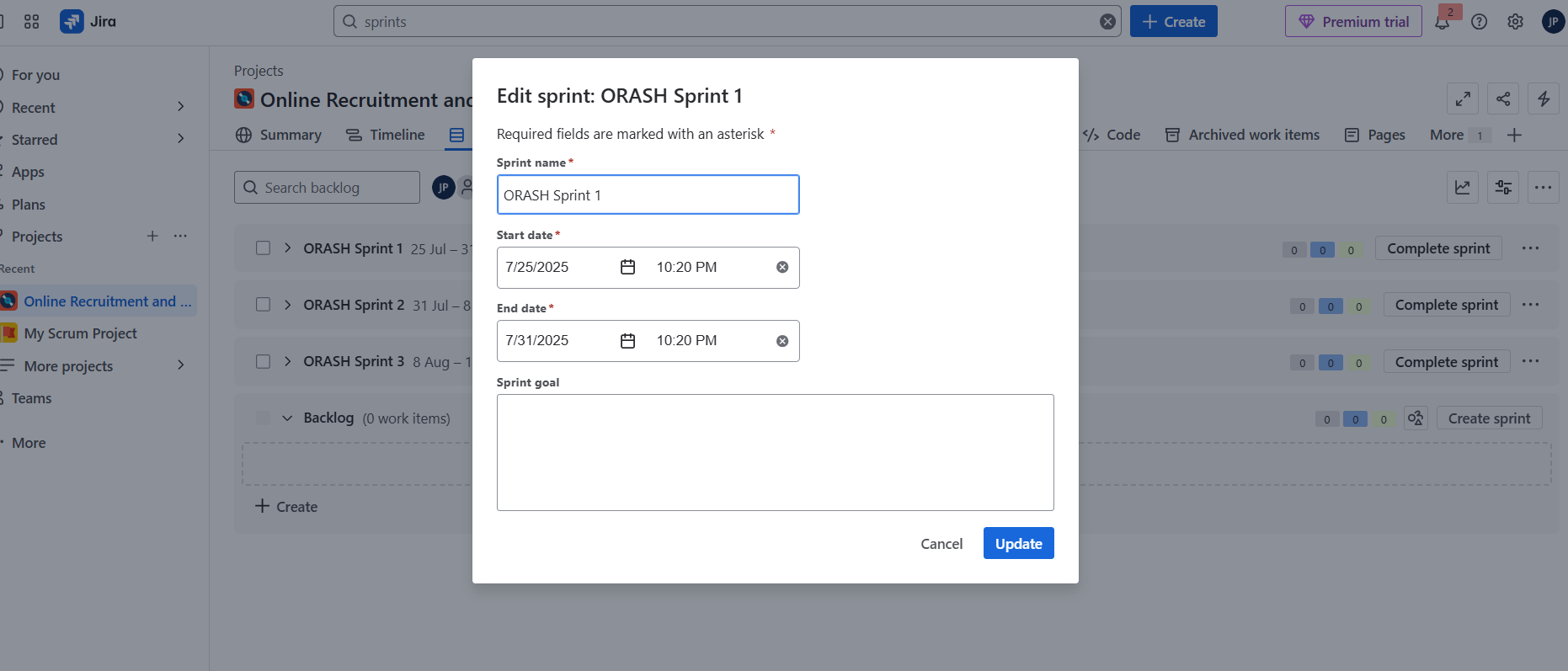


Figure Showing the sprint 1 start date

## 3.7 Provide a Screenshot of Jira Board

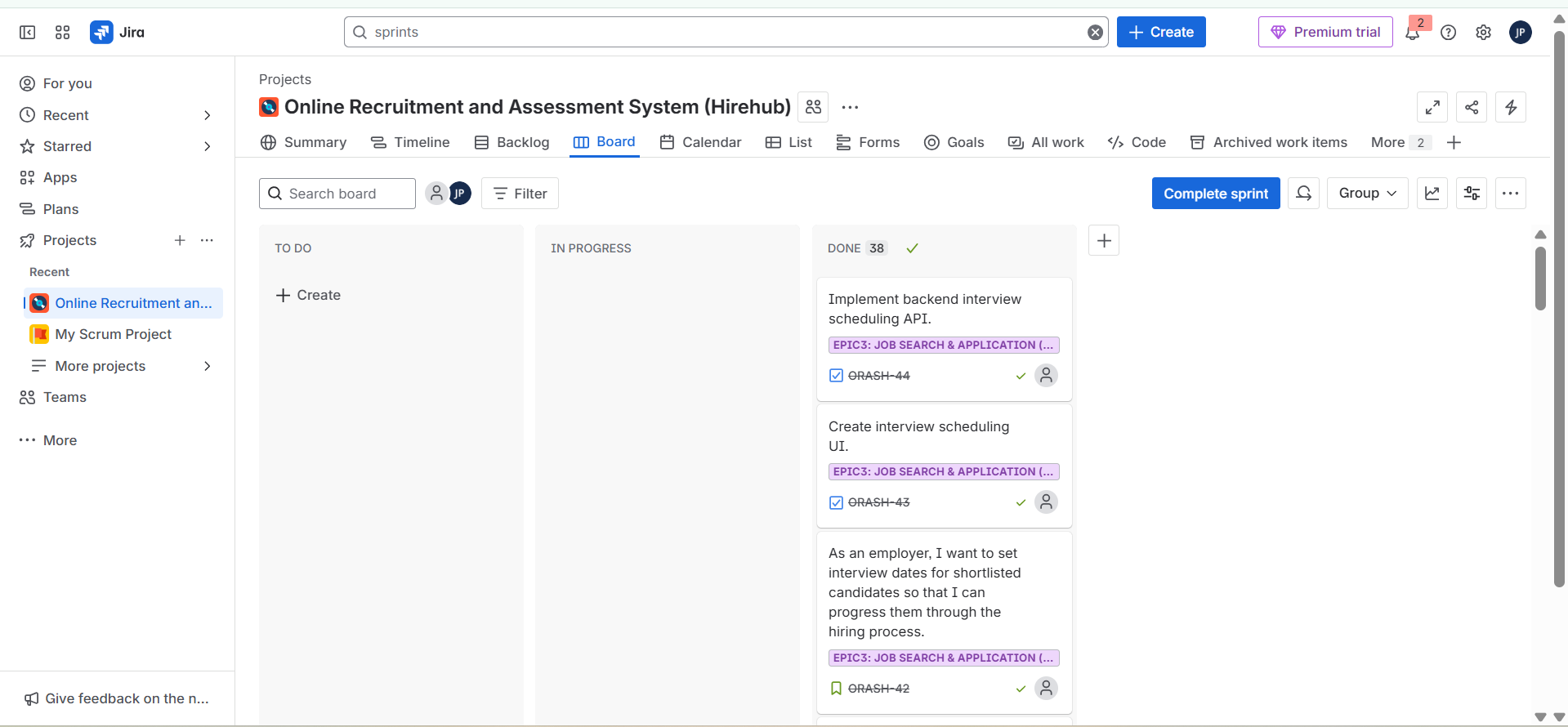


Figure Jira board Screenshot

## 3.8 Provide a Screenshot of the completed sprint

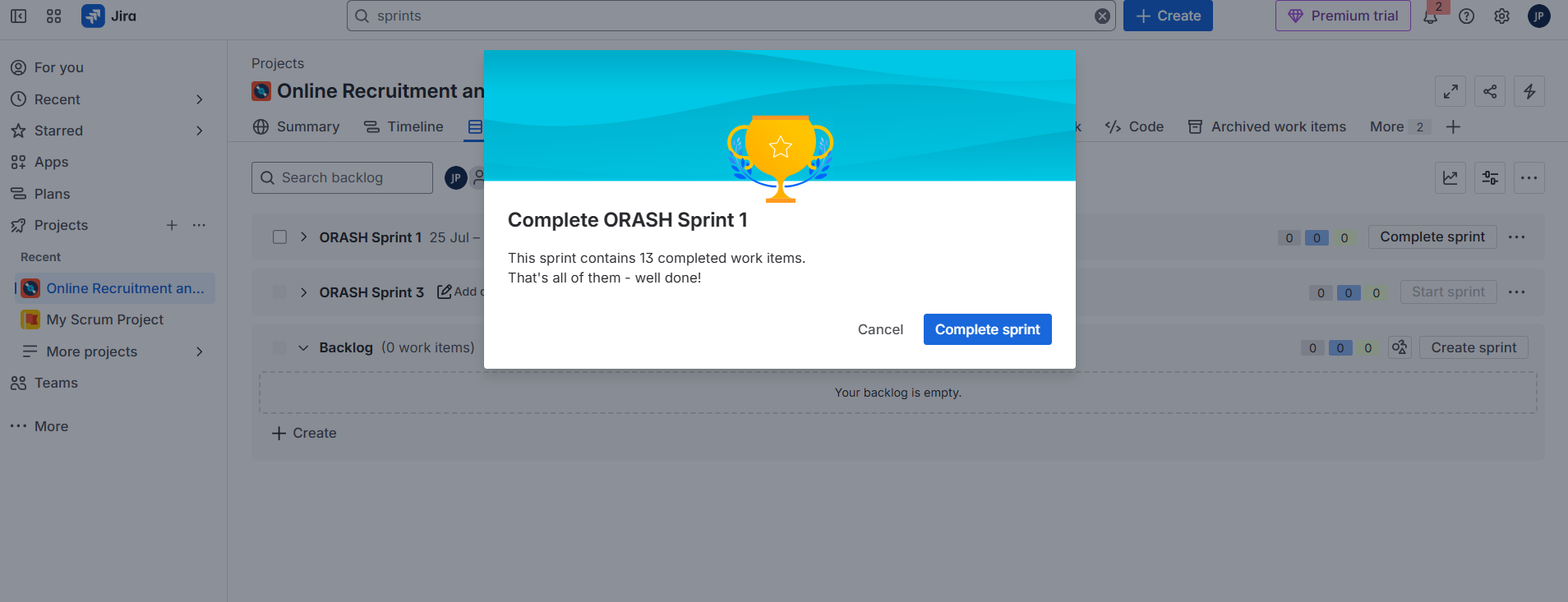


Figure Completed all sprints.

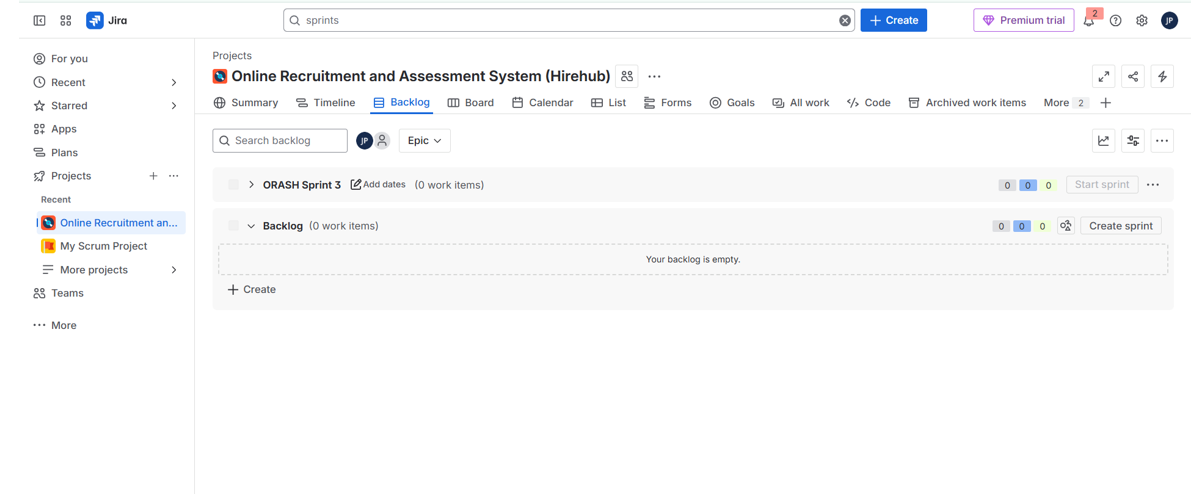


Figure : No backlog after all sprint completion

## 3.9 Provide a Screenshot of the commit history from JIRA

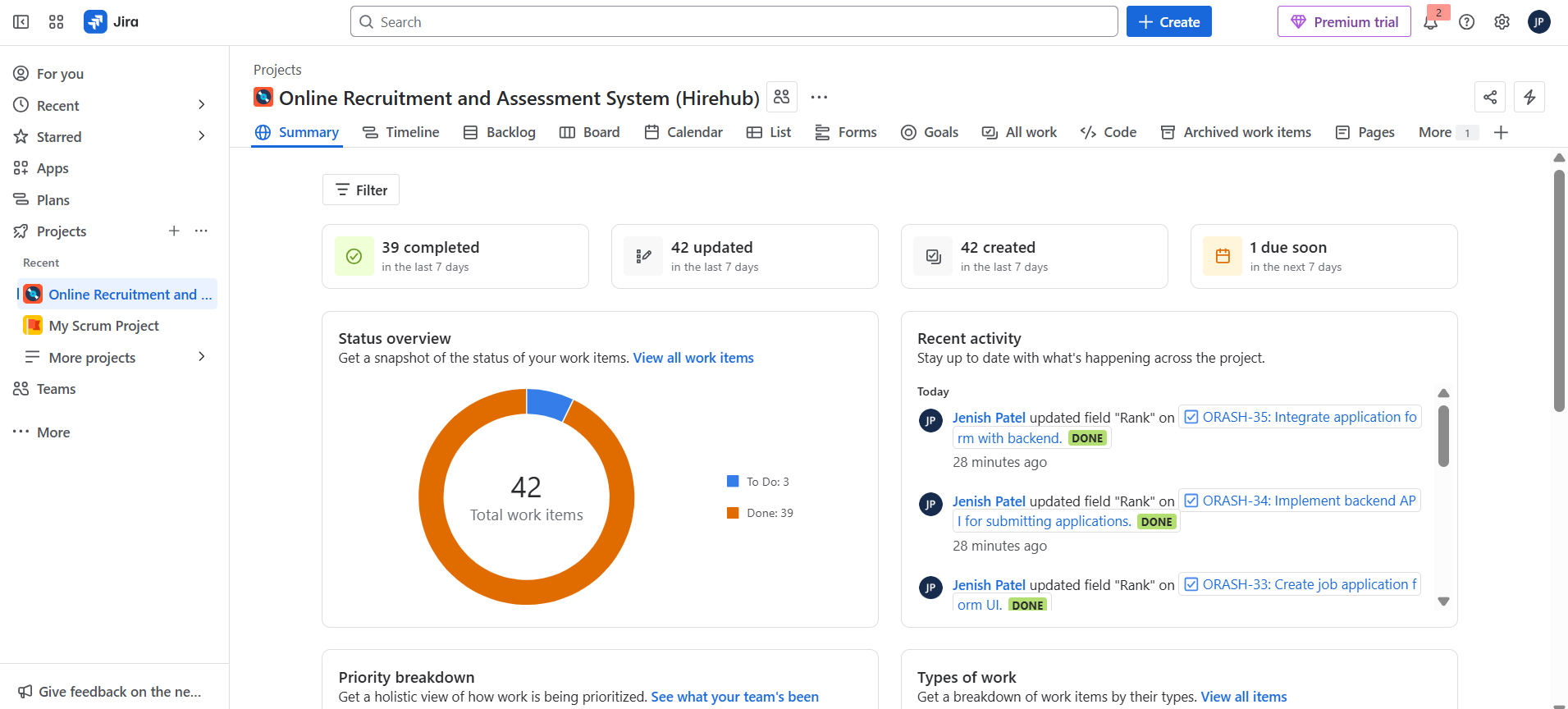


Figure : Sprint Summary

# Backend Development, Frontend Development, GitHub Version Control & Branching Strategy

*Just provide your GitHub Link and public URL (****your instance’s public IP address)****. We will review your code implementation to verify whether the backend is functioning correctly through the public URL. Additionally, we will also review your commits, main branch, feature branches, and pull requests.*

Github URL: <https://github.com/Jenish2304/Online-Recruitment-and-Assessment-System.git>

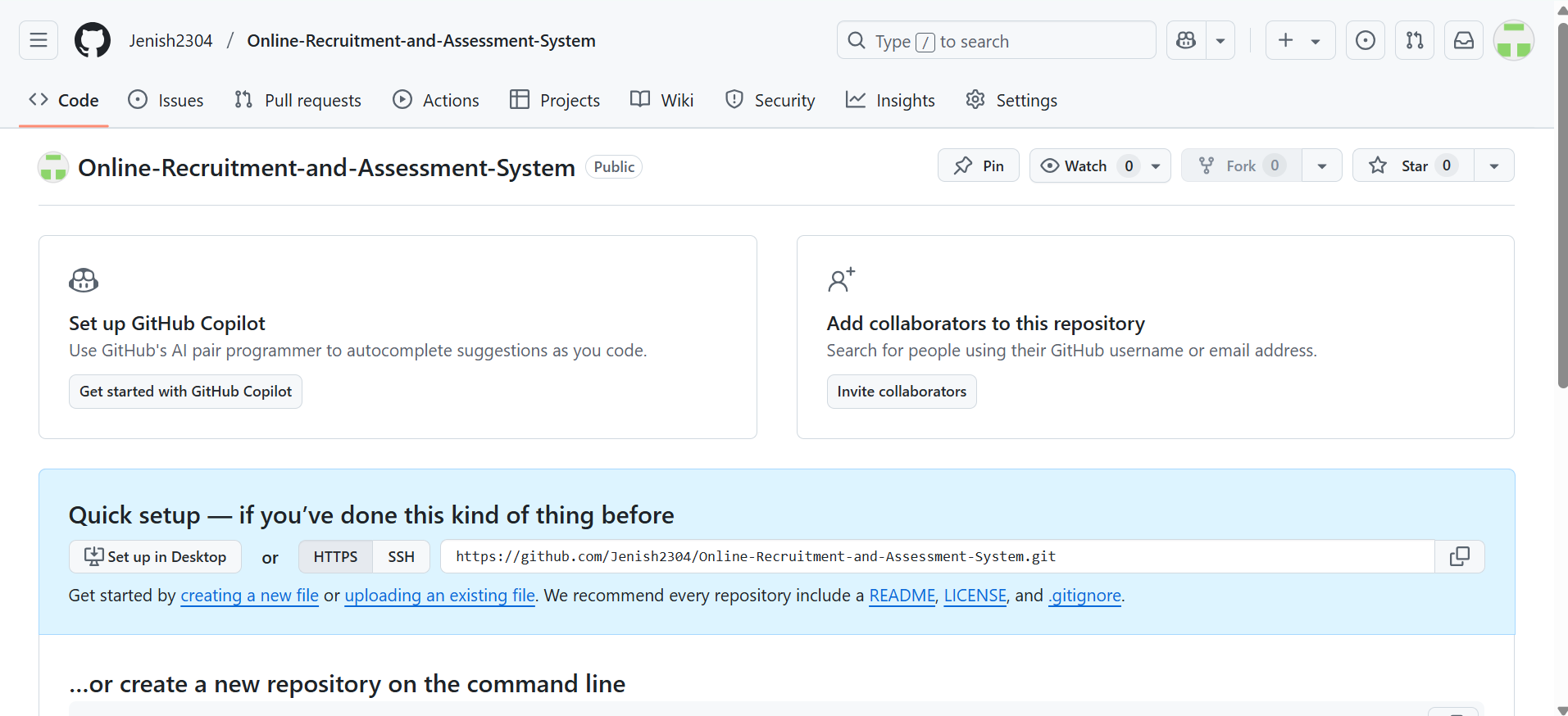


Figure Created Public Repo

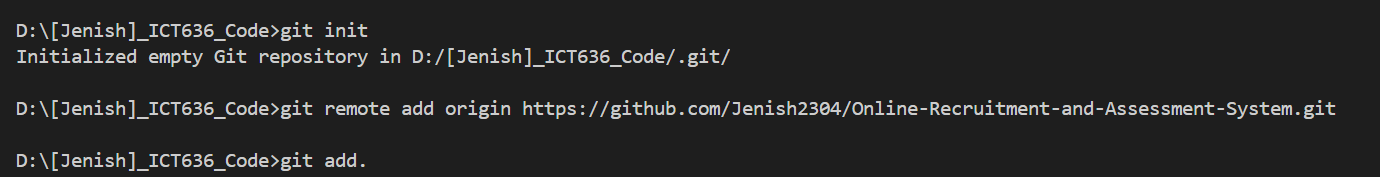


Figure initialised the repo.

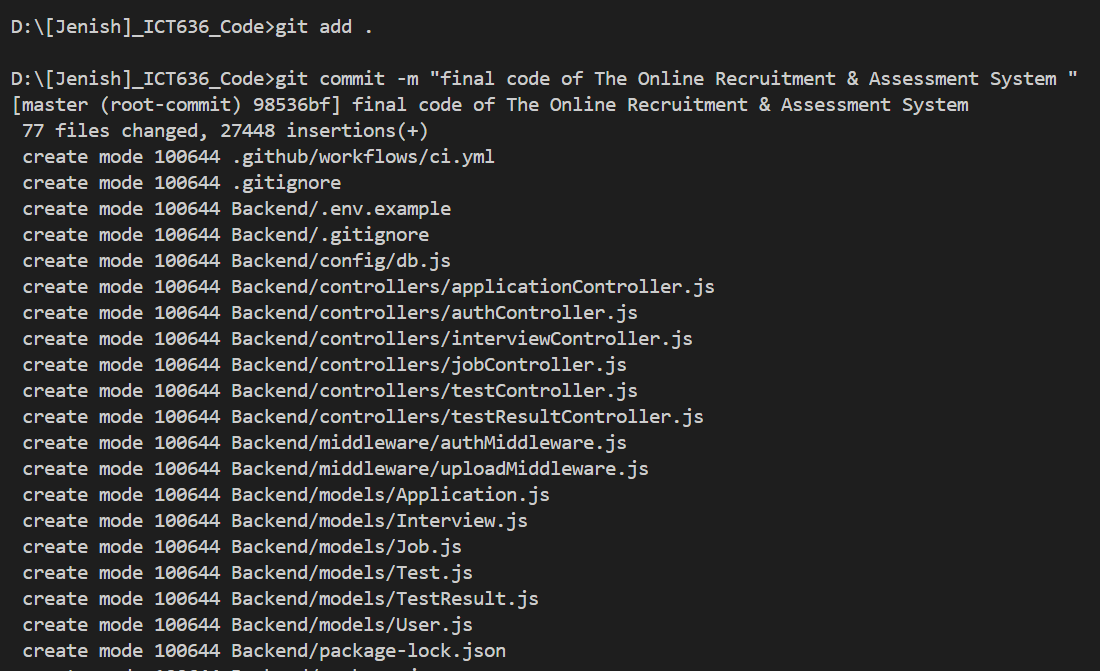


Figure added all the code.

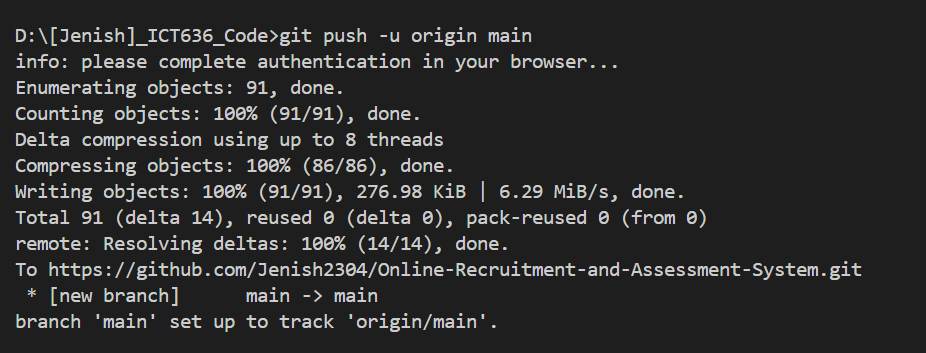


Figure : commit and pushed code on github

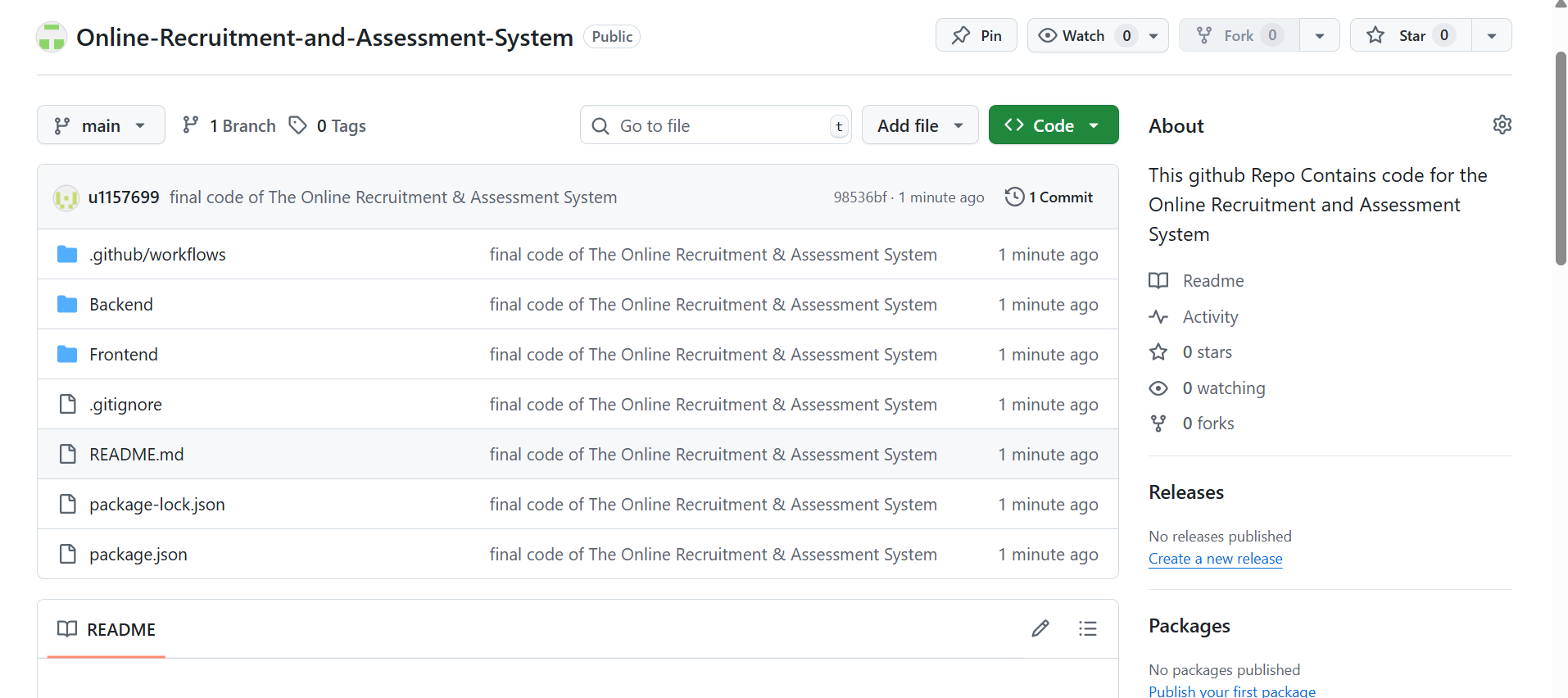


Figure Final code Pushed On github

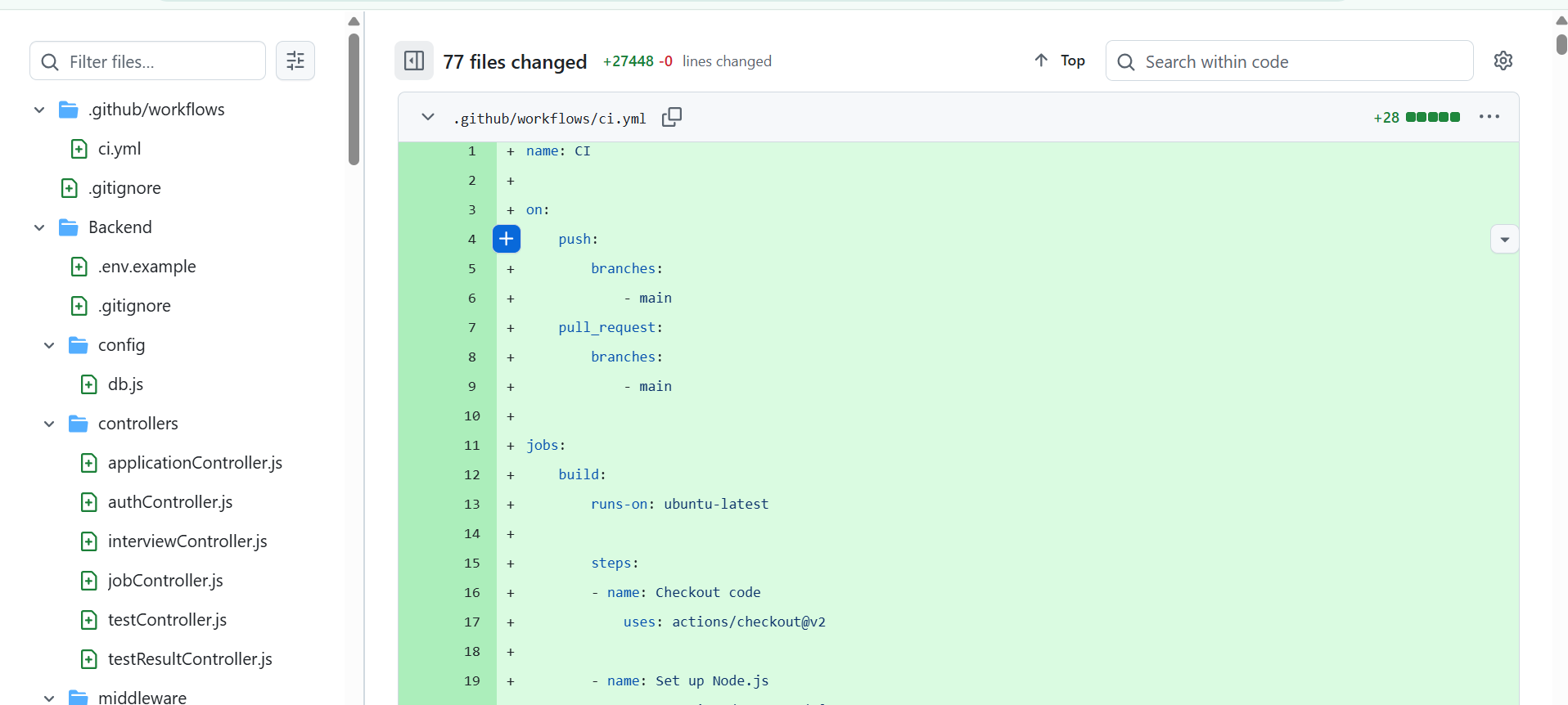


Figure : Open a file on GitHub

# CI/CD Pipeline Setup

## 5.1 Provide YML file screenshot

## 5.2 Provide a Test Case pass/fail status screenshot (from the terminal output)

## 5.3 Provide a GitHub Action Configuration screenshot (Include runner, environments, and prod variables setup)

## 5.4 Provide a screenshot of EC2 server configuration (Only include pm2 status output)

## 5.5 Provide a screenshot of the running project, including public IP (Only one screenshot)

## 5.6 Provide a screenshot of the Final Workflow Run Test through GitHub Action (where the job is running when you push, and you can see all of them are passing or something fails)

# README.md (you can include GitHub readme.md file link or screenshot)

# Discussion and Conclusion

The total software development lifecycle (software requirement analysis to deployment via DevOps practices) was an effective show in the development of the HireHub Job Platform. In this project, we included the backend and frontend in one project, and it was designed in a modular and scalable way. Backend was developed based on Node.js, Express, MongoDB and centred on responsibilities like authentication, job posting, applications, screening test and interview scheduling. The frontend, which was aimed at creating user user-friendly interface using React.js, had role-based access to candidates and employers.

The SysML similarity requirement, block definition and parametric diagrams significantly defined the functional and non-functional requirements, cases associated with component relationships and limitations so that the design decisions were properly crafted. JIRA was effectively adopted in project management, and it allowed working on sprint planning, backlog and user story management. Version control was done in GitHub, and it had a branching plan to keep the code steady.

The CI/CD pipeline based on GitHub Actions and initiated on AWS EC2 with PM2 allowed automating the build, testing, and deployment processes, as well as facilitated the successful integration of all product parts and minimised manual efforts.

Altogether, the project accomplished its goals as it produced a working application that was deployed to the cloud. Not only was it an introduction to full-stack development on a technical level, but it also instilled a sense of the importance of controlled project management, continuous integration/deployments in the working processes of actual software engineering as well. What is obtained at the end is a well-tested, supportable platform that can be used in practice.

# Reflection

The experience of working on the HireHub platform has been a highly intensive course of learning, including a technical part and a project management part of software engineering.

One of the things I understood at the requirements stage was the importance of collecting and organising system requirements, and visualising requirements prior to coding. The development of the SysML diagrams was a handy practice of documenting requirements, component connections and constraints. It also guided me to become a more systematic thinker and anticipate possible integration concerns at the very beginning.

During the backend development process, I further got acquainted with the knowledge of designing RESTful APIs and secure authentication with JWT and cookies. Making candidates and this method of access control by reliable jobs was initially a hard task, but became easier as I designed middleware to control authorisation. Data persistence using MongoDB also enabled me to learn how to model real-life entities such as jobs, applications and interviews.

The React.js frontend development helped me build expertise in UI development based on components and state management capability, as well as applying APIs. My knowledge of the importance of responsive design and consistent styling in developing a good user experience also increased.

JIRA was quite instrumental in the tracking and task management. The division of features into epics, user stories, and subtasks enhanced the level of focus and clarity on sprints. The work to update the board regularly resembled industry-based agile development processes.

One of the greatest lessons learnt was the implementation of the CI/CD pipeline. Using GitHub actions to automate builds, testing and deployment to an AWS EC2 instance taught me the importance of the DevOps concept in the development process. This not only saved on conventional work but also raised confidence in the code stability.

Nevertheless, it was not smooth sailing. I experienced problems with CORS debugging between the front and back ends, cookie-handling security, and AWS EC2 configuration of the environment to work in production. Scheduling of JIRA sprints with real-life development needed to be changed sometimes. It was also a steep learning curve to learn GitHub Actions YAML configuration, but these challenges resulted in some considerable improvement in my DevOps skills.

To sum it up, the given project provided an extensive practical experience of full-stack application development, team management, agile project management, and continuous delivery. The synergy of both technical implementation and designing documentation, in addition to cloud deployment, has also made me confident in handling future end-to-end software projects.

# References:

Bruckner, R. (2023). *New demo: SysML Parametric Diagram (PAR)*. Jointjs.com. https://www.jointjs.com/blog/demo-wednesday-parametric-diagram

VisualParadigm. (2023, September 21). *Beginner’s Guide to Requirement Diagrams in SysML - Visual Paradigm Blog*. Visual Paradigm Blog. https://blog.visual-paradigm.com/beginners-guide-to-requirement-diagrams-in-sysml/

Walker, W. (2025, January 31). *SysML Block Definition Diagram - CameoMagic*. CameoMagic. https://cameomagic.com/sysml-block-definition-diagram/