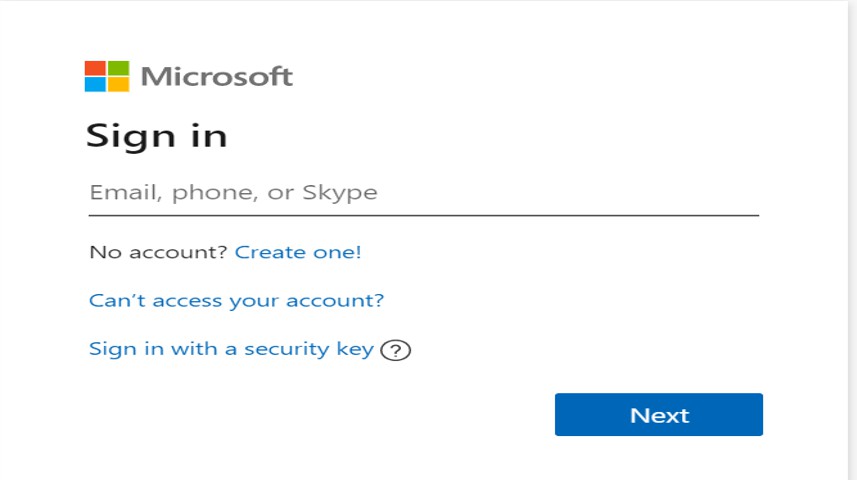
Ex. No:1 Setting up a DevOps Team with a simple project

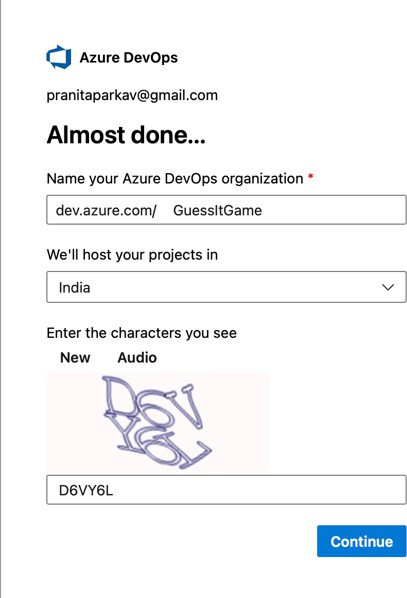
## Date :

**Aim :**

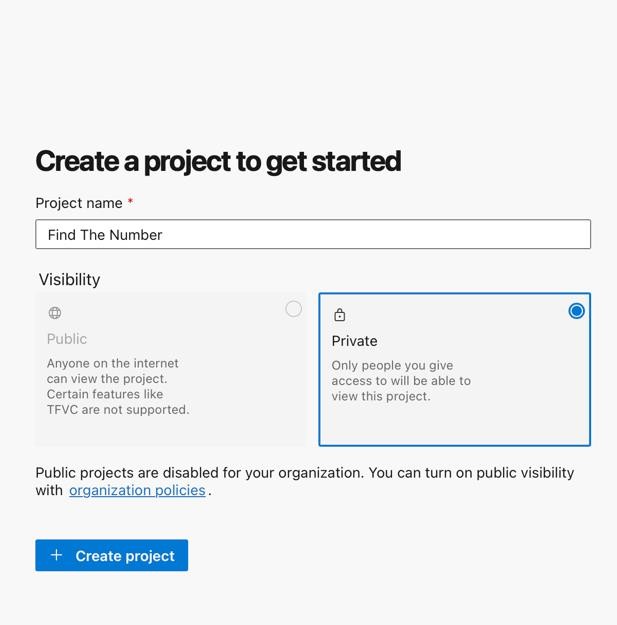
To create a DevOps Team with a simple project and explore the Devops environment.

## Procedure:

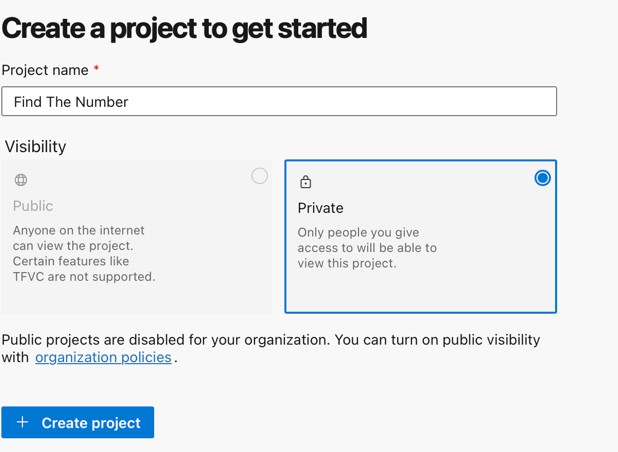
1. Create a microsot oulook mail id if you do not have already.
2. Sign in to the azure devops environment. (dev.azure.com)
3. Create devops organization.

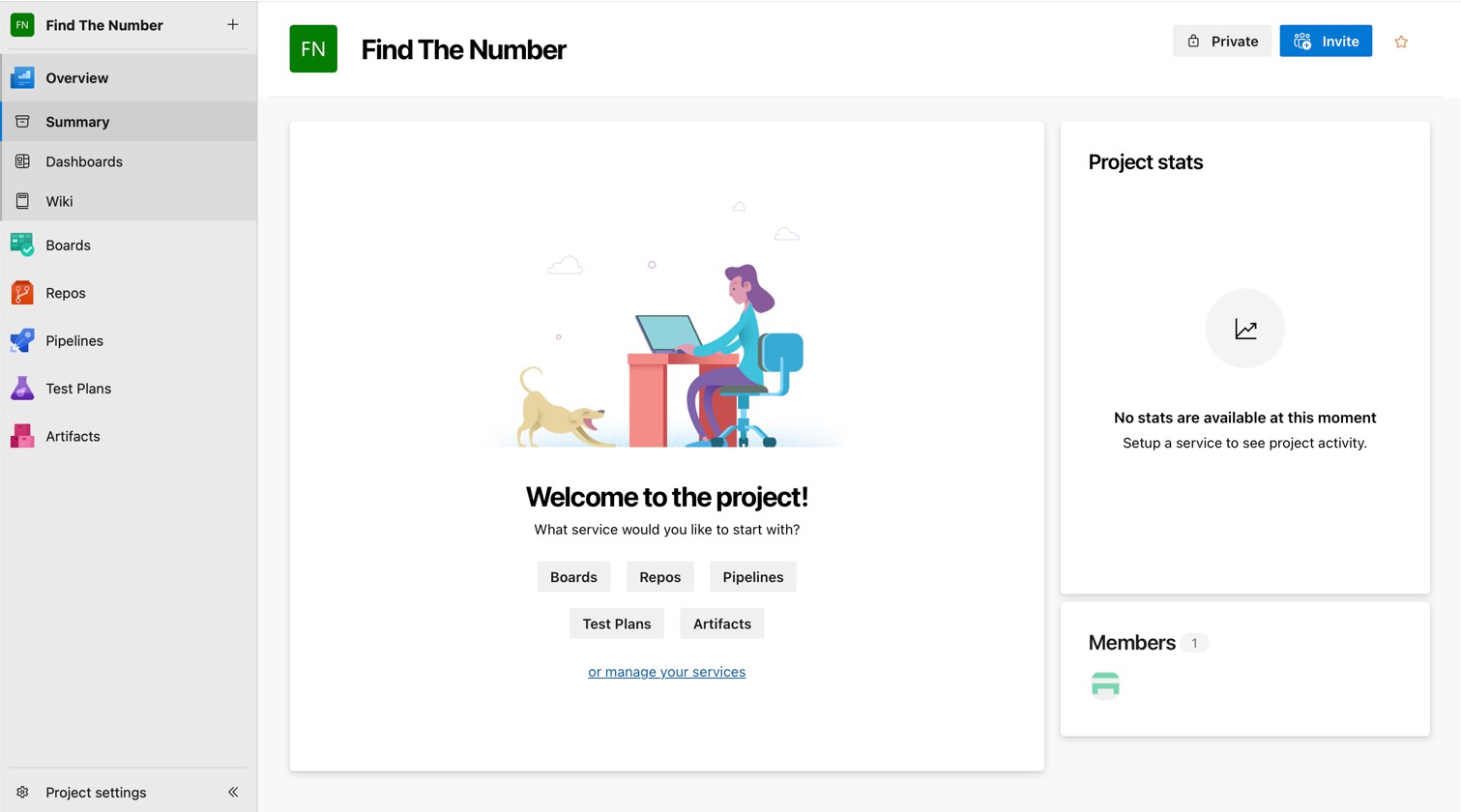


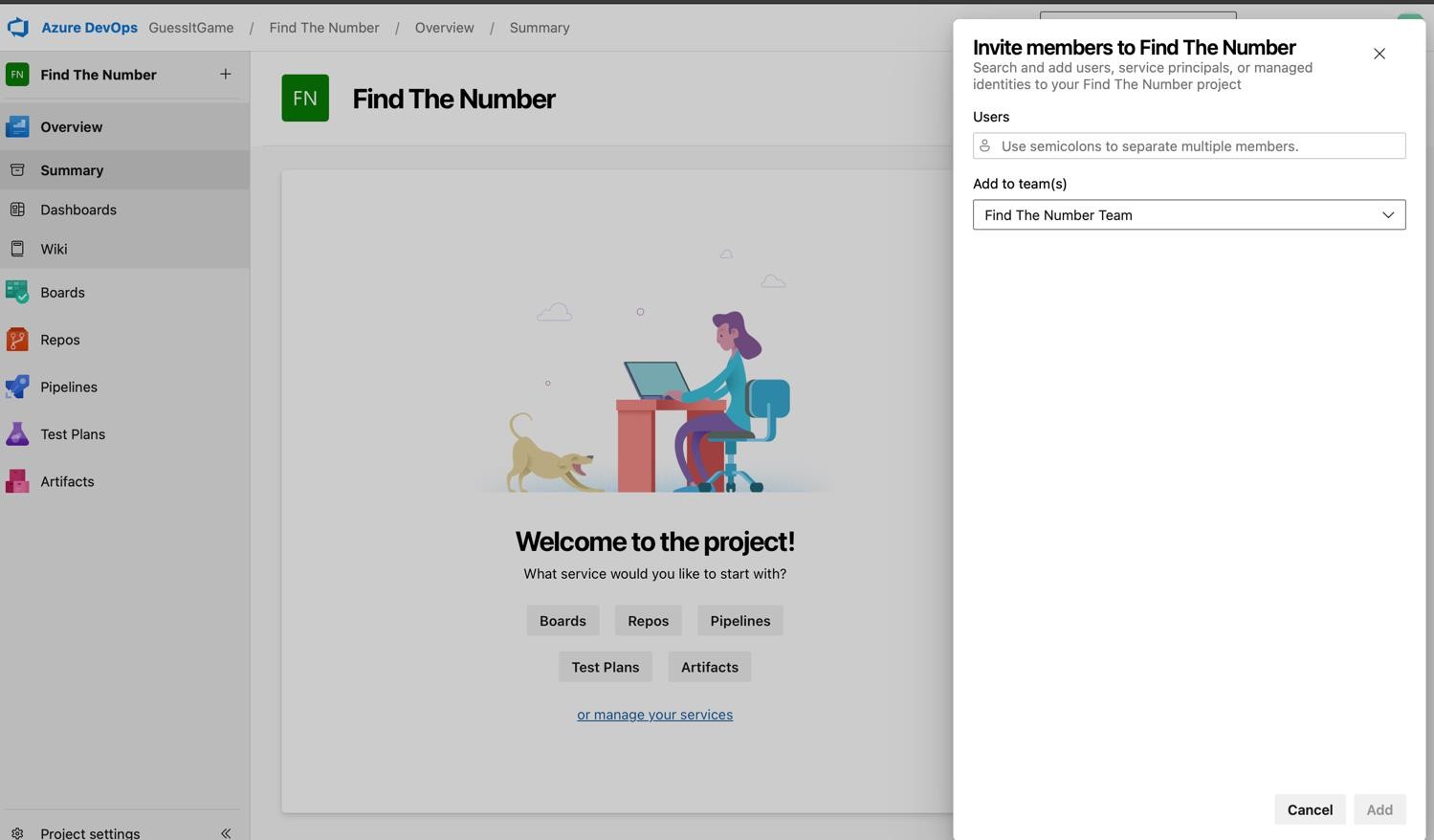
Create Project with Basic Process in Azure DevOps by clicking on the New Project button on the top right corner.



1. Choose the possible visibility level.



1. Click **Create** button. Navigated to Welcome page.
2. Choose the options display on the welcome screen to continue.
3. **Invite:** Using this option you can add others to your project. You can only invite users who are already part of your organization.



## Result :

Thus the simple project was created in azure devops.

**Ex. No:2 Creating and manage a SPRINT backlog and tasks, and track progress with boards**

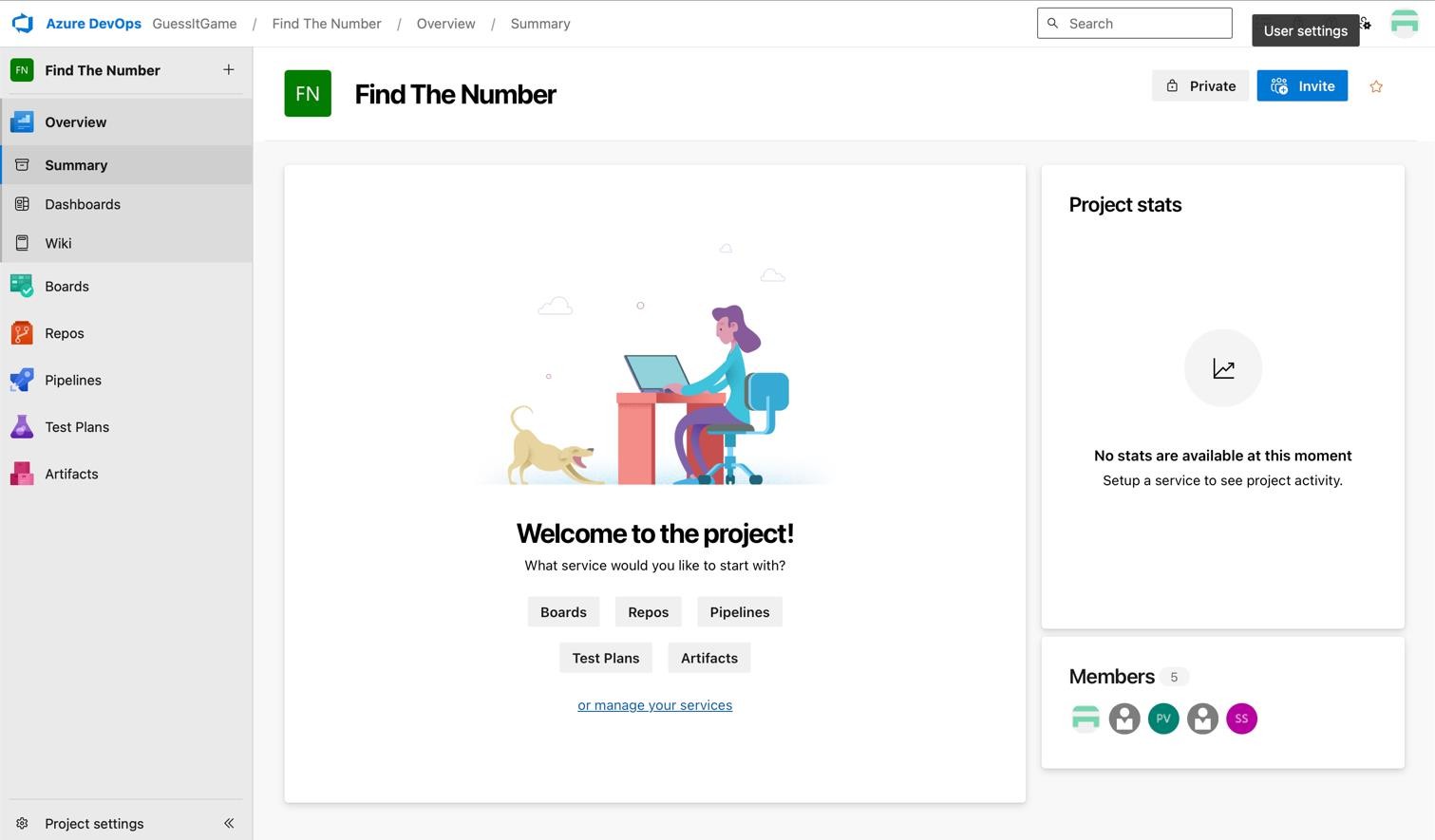
**Date :**

**Aim :**

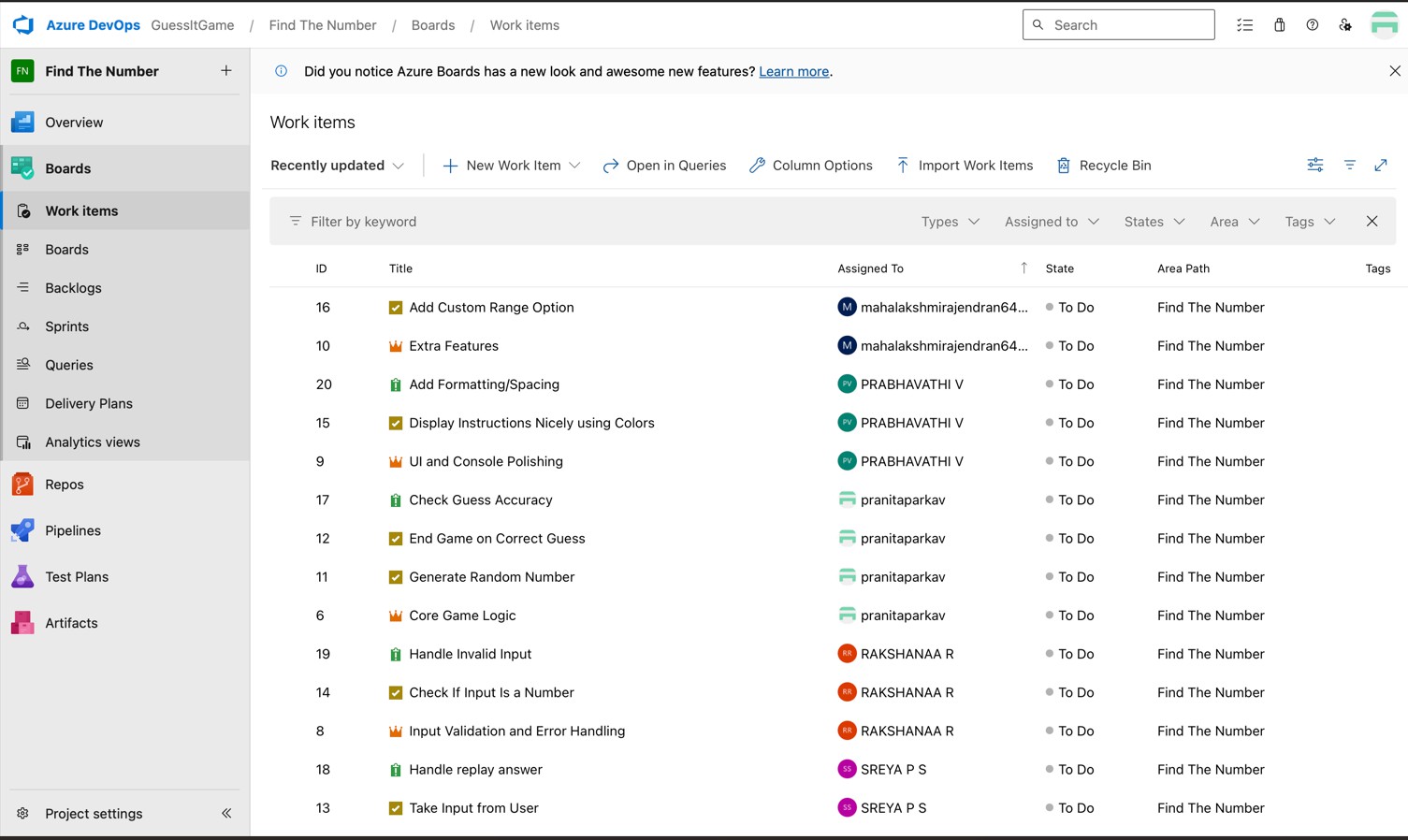
To manage Sprint tasks and track progress using boards.

**Procedure:**

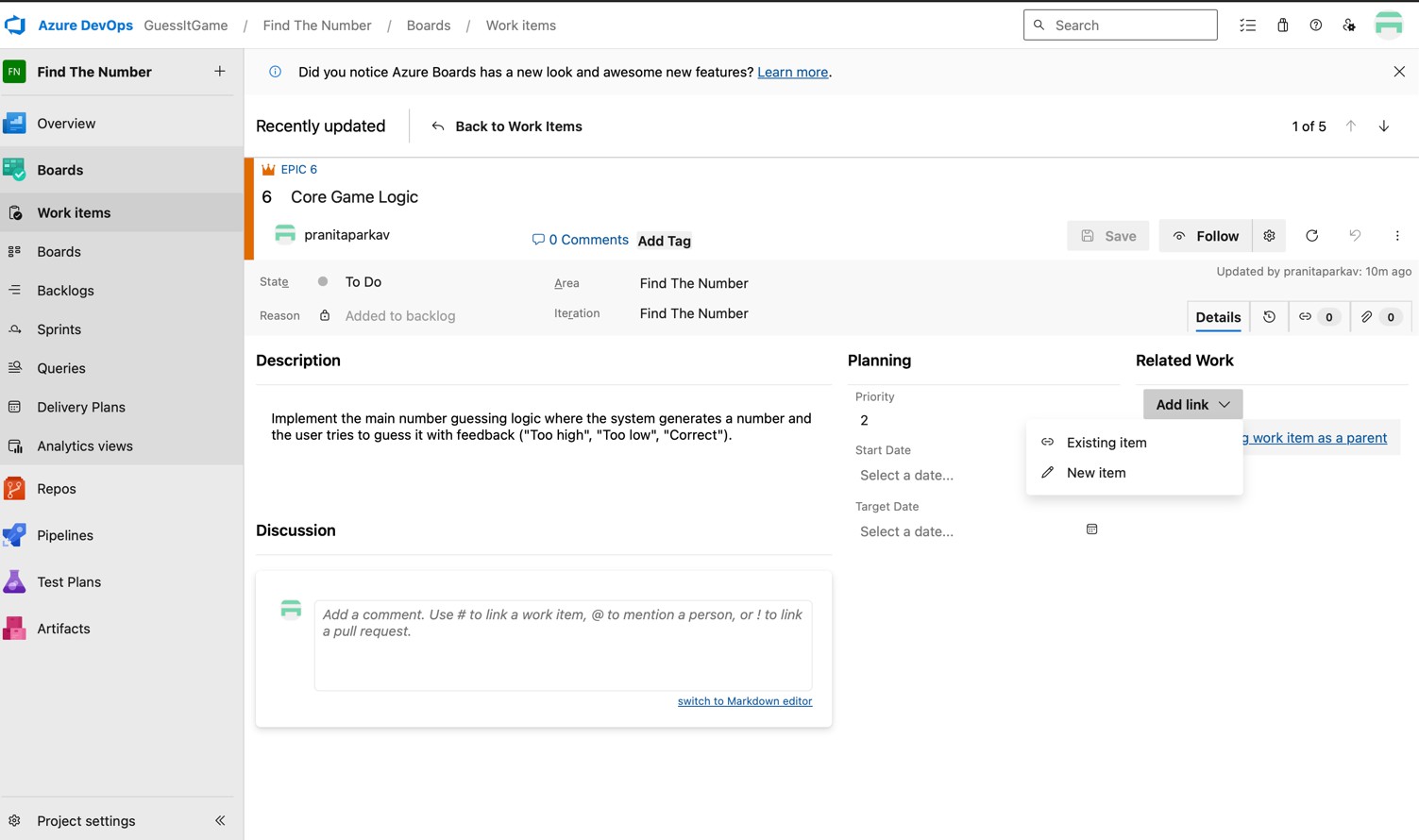
1. Dividing the team members into the design team and the development team.



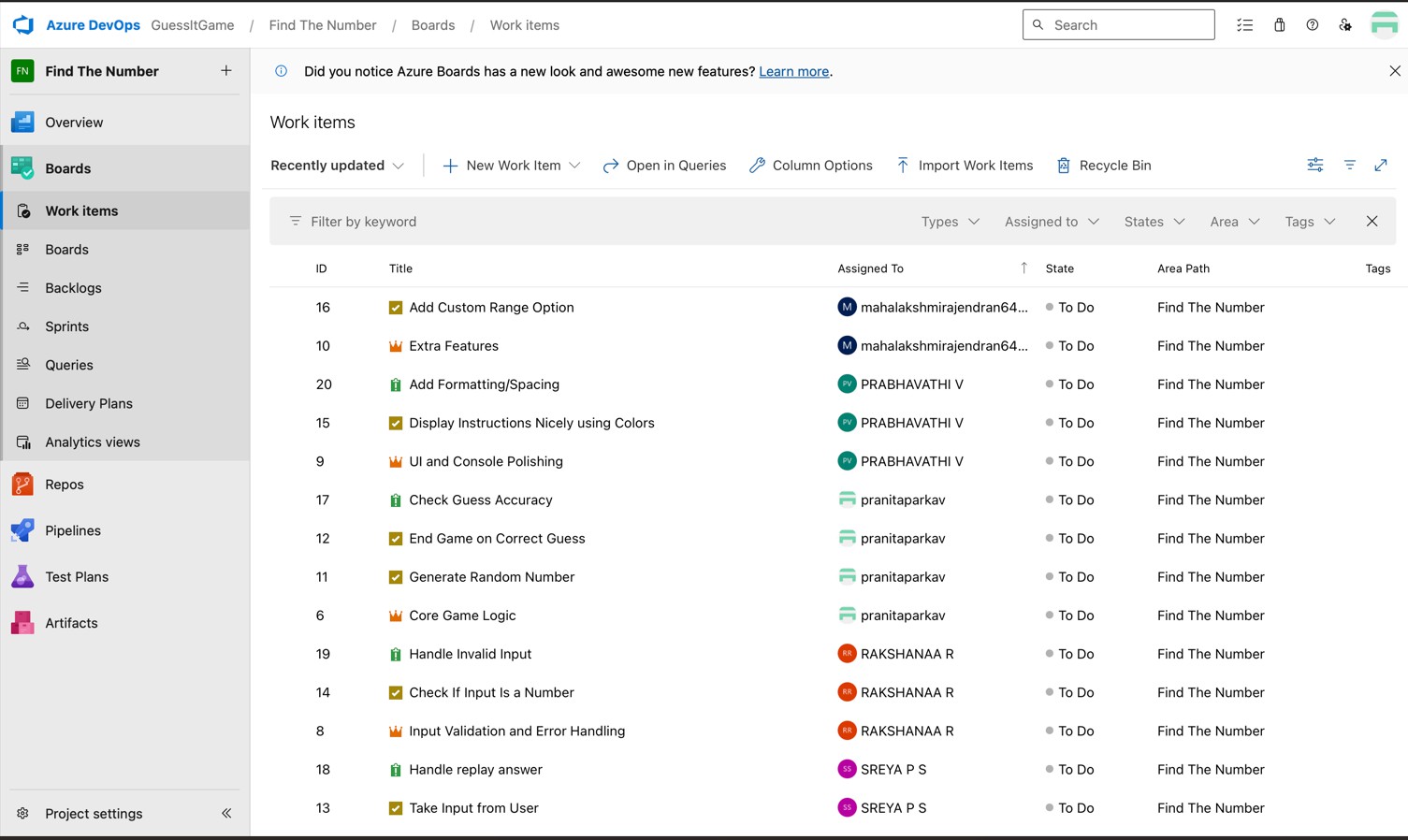
1. Adding the sprints and backlogs.



1. Assigning the task and writing the user story in description.



1. After assigning tasks, the work items are allocated to specific team members.



## Result :

Thus, the Sprint backlog was created, tasks managed, and progress tracked using boards.

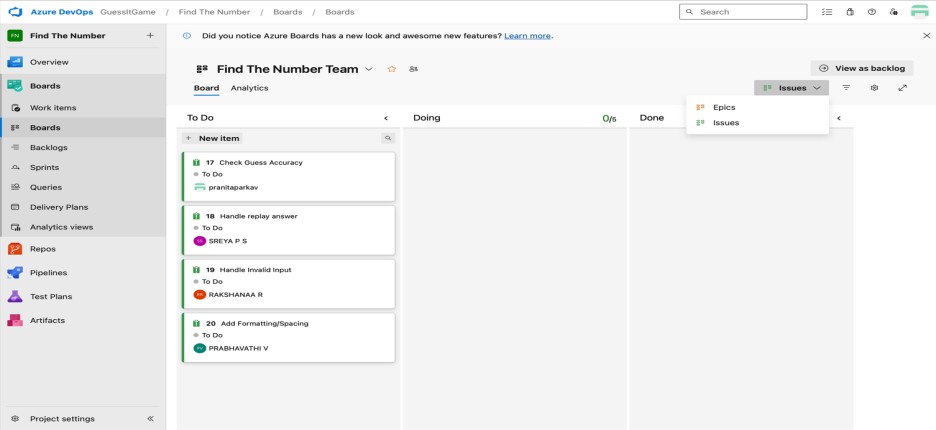
**Ex.No: 3 Set up a KANBAN board and establish work-in-progress limits. Date :**

**Aim :**

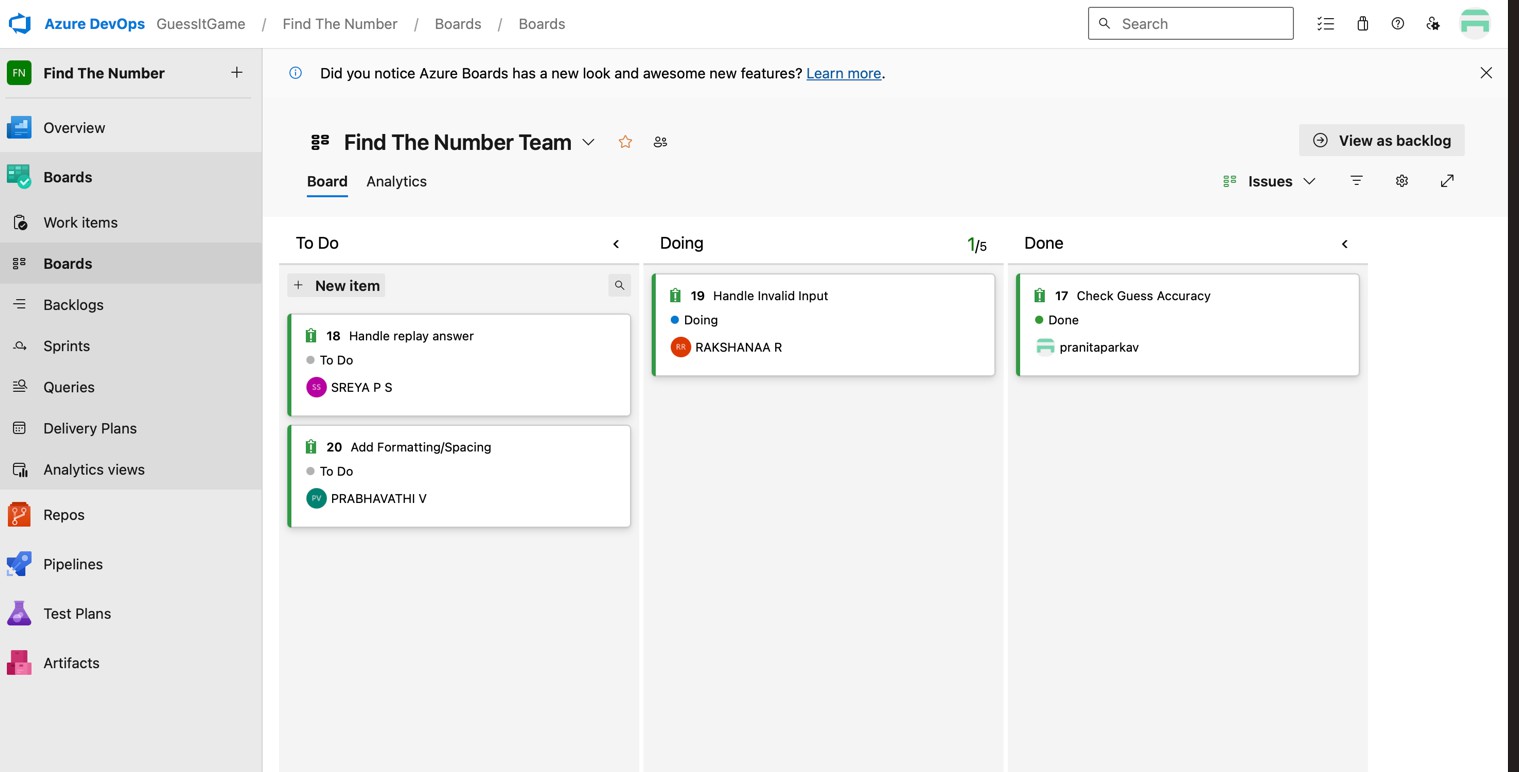
To set up a Kanban board and limit tasks in progress for better workflow.

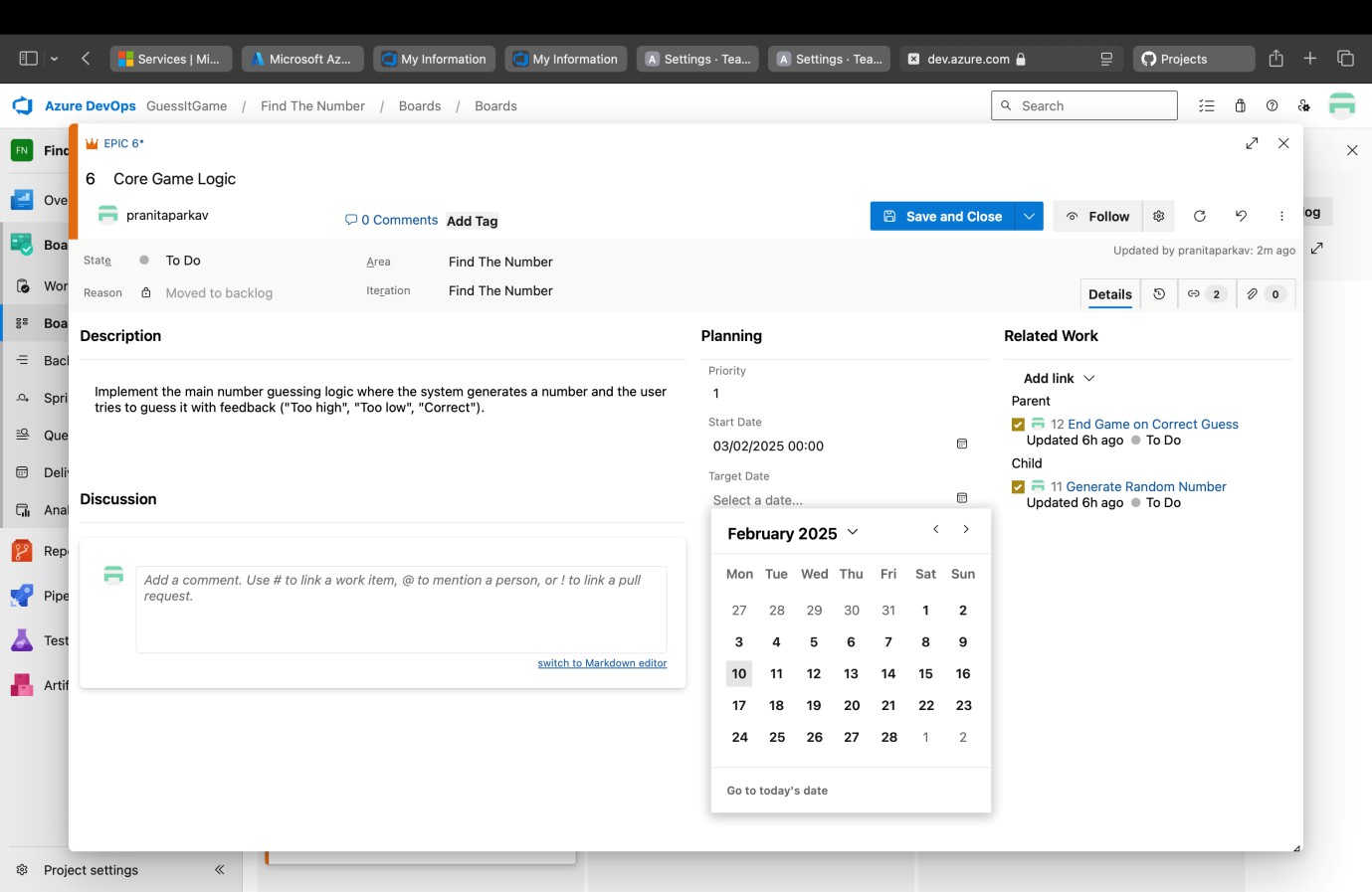
**Procedure:**

1. Using the boards to track the progress (To Do, Doing, Done).

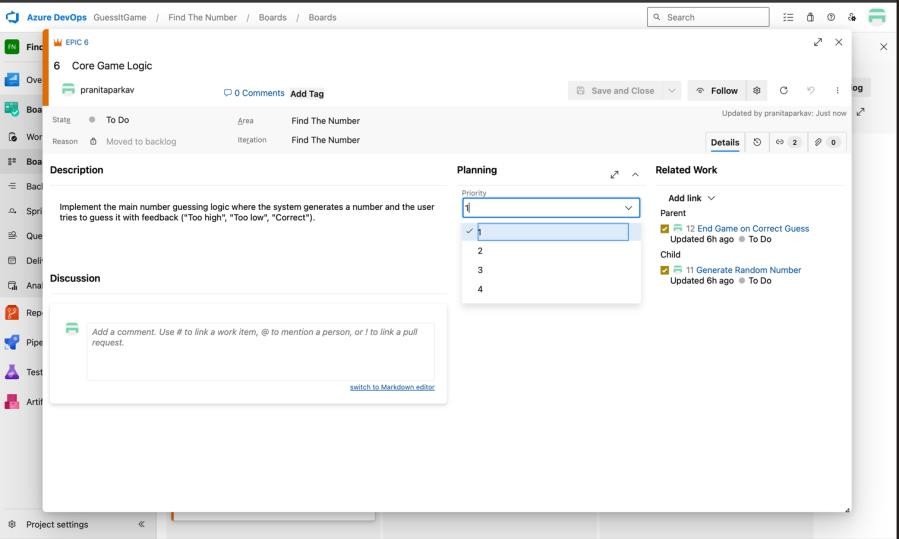


1. Using the boards to track the progress (To Do, Doing, Done).

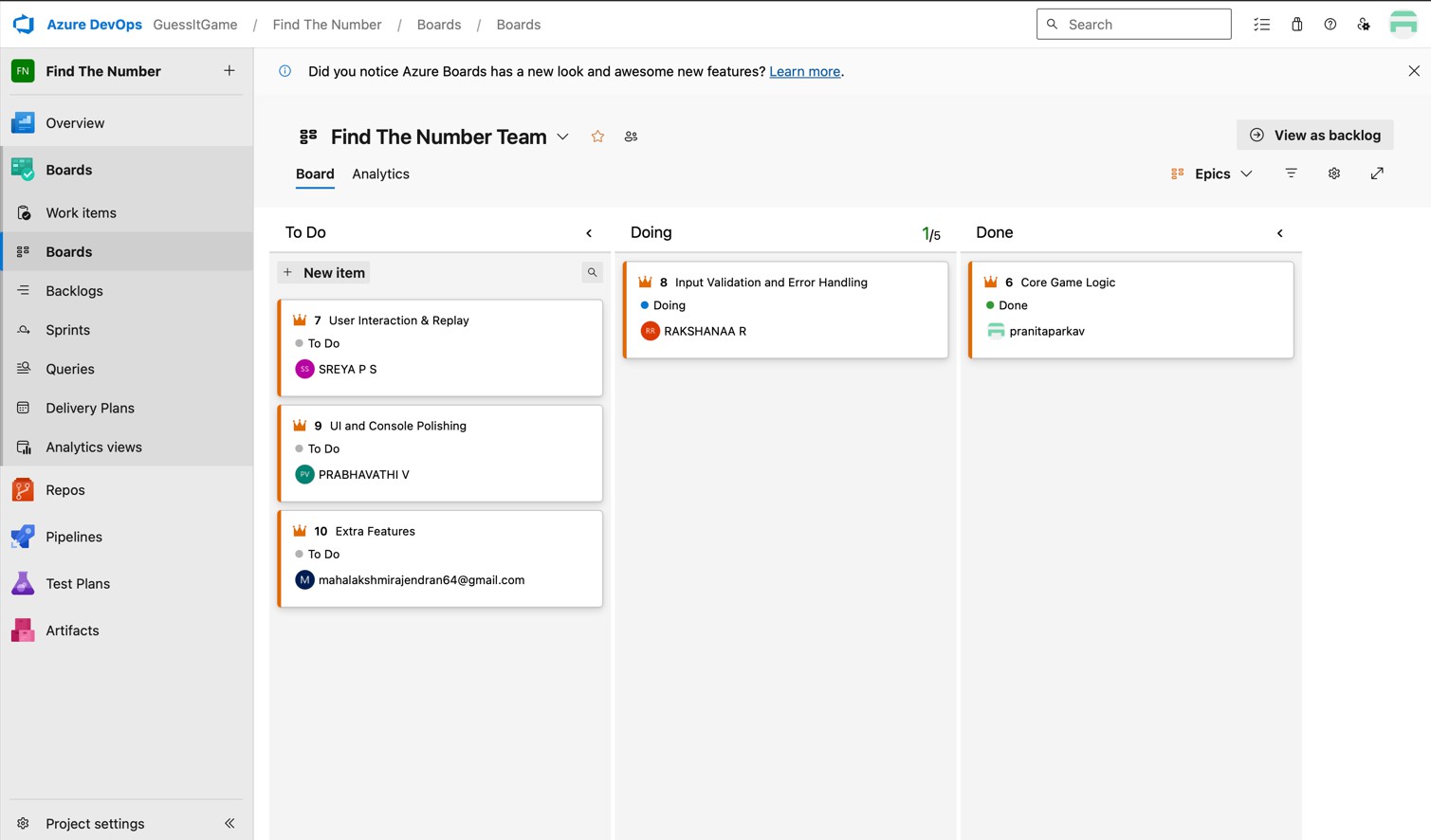




1. We can set priority for a particular task to give it importance.



1. In Boards, individual tasks can also be prioritized and managed under categories such as To Do, Doing, and Done.



## Result :

Thus, a Kanban board was set up, and work-in-progress limits improved task flow.

**Ex.No: 4 Create a new repository, make changes, and commit them for version control.**

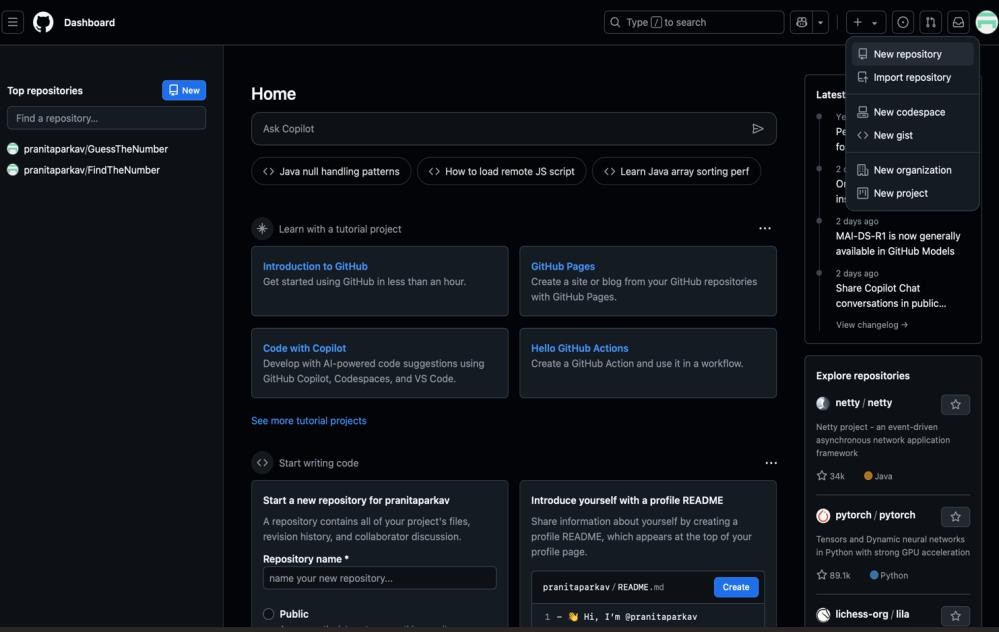
**Date :**

**Aim :**

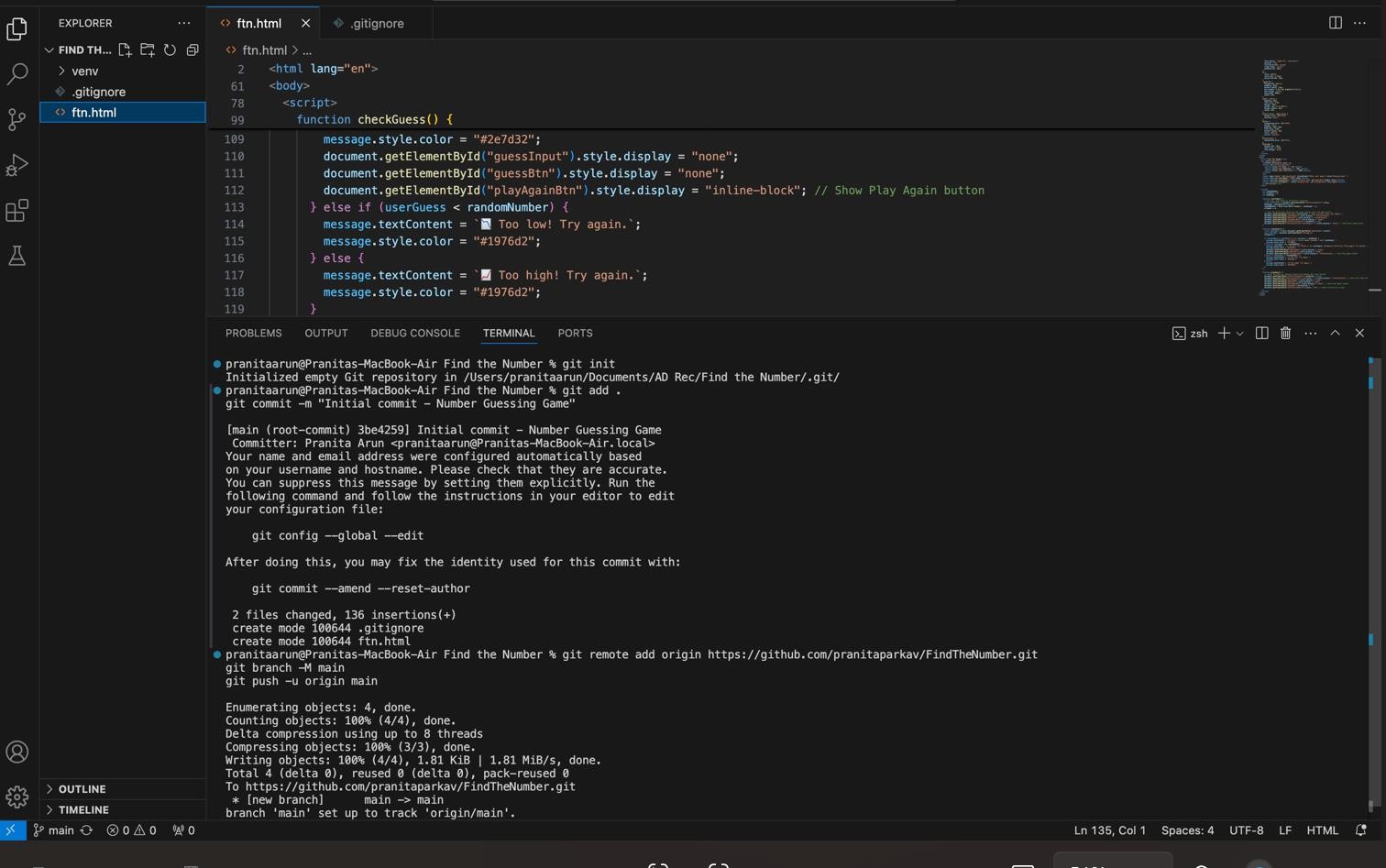
To create a repository, make changes, and commit them for effective version control.

**Procedure:**

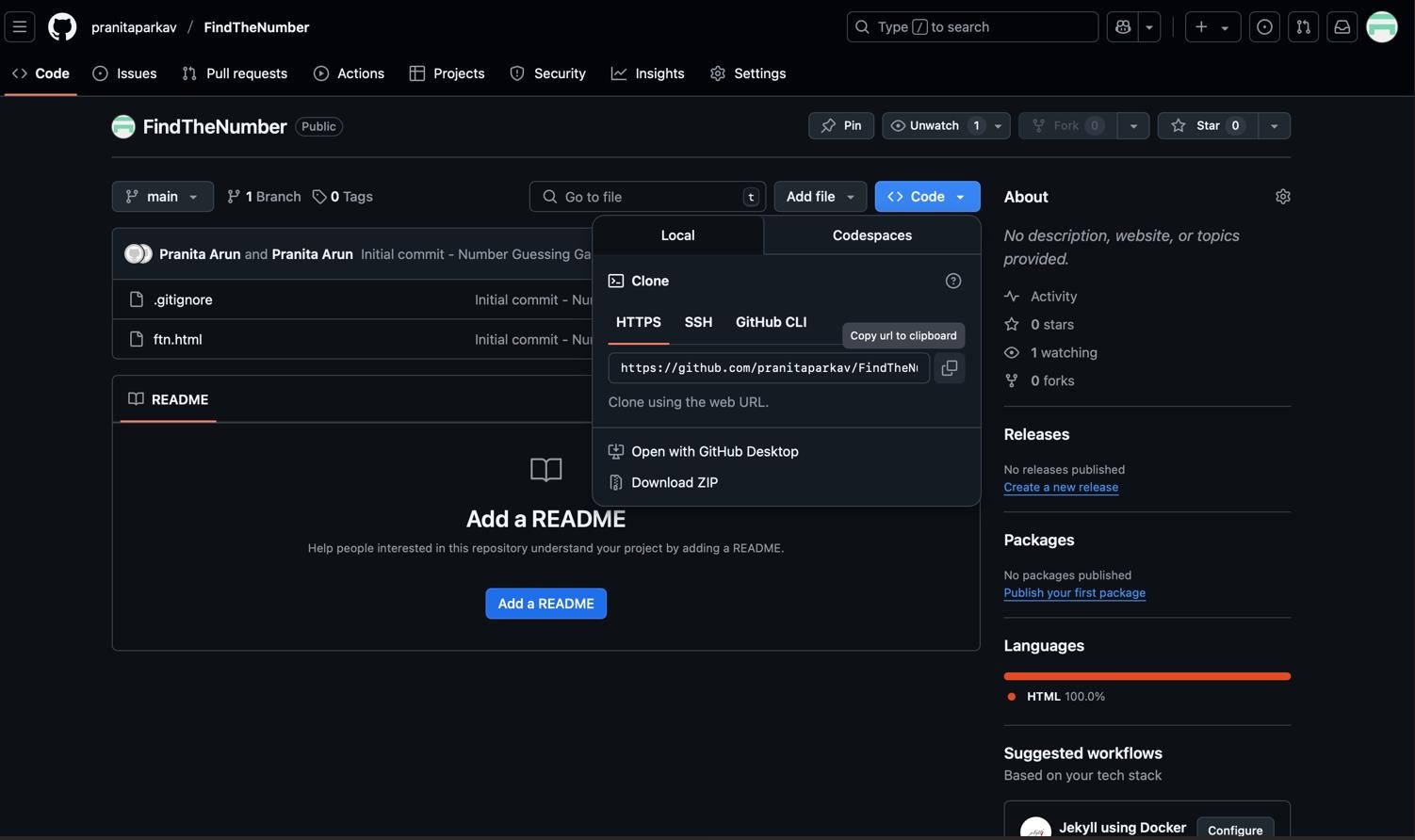
1. Log in to GitHub.
2. Creating a new repository.



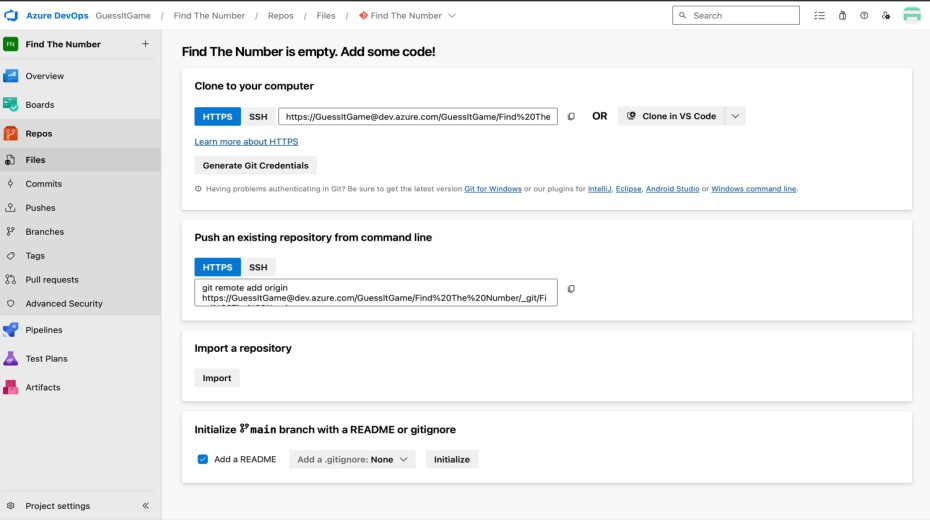
1. Adding the code to the repository.



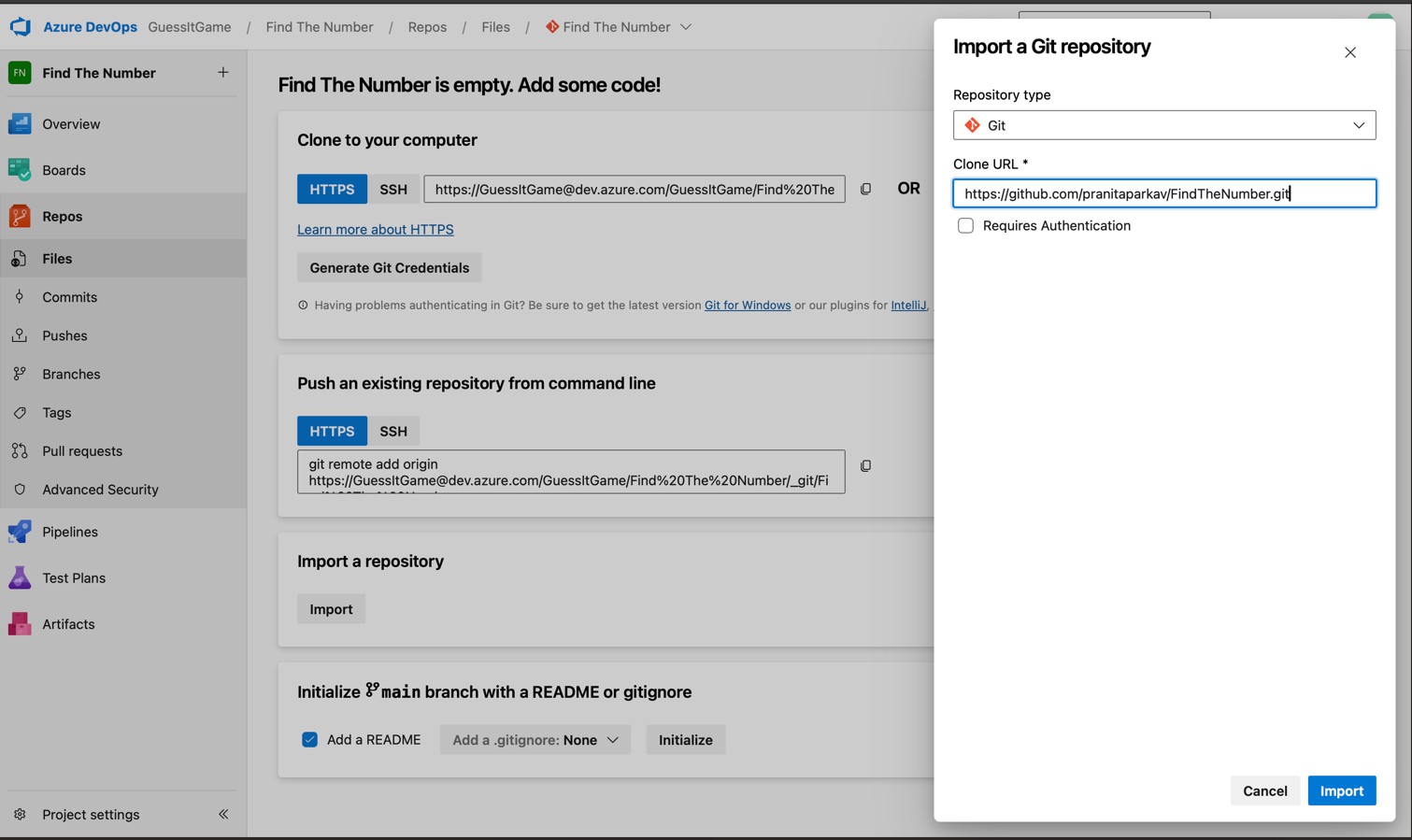
1. Copying the URL of the repository to export to Azure DevOps.



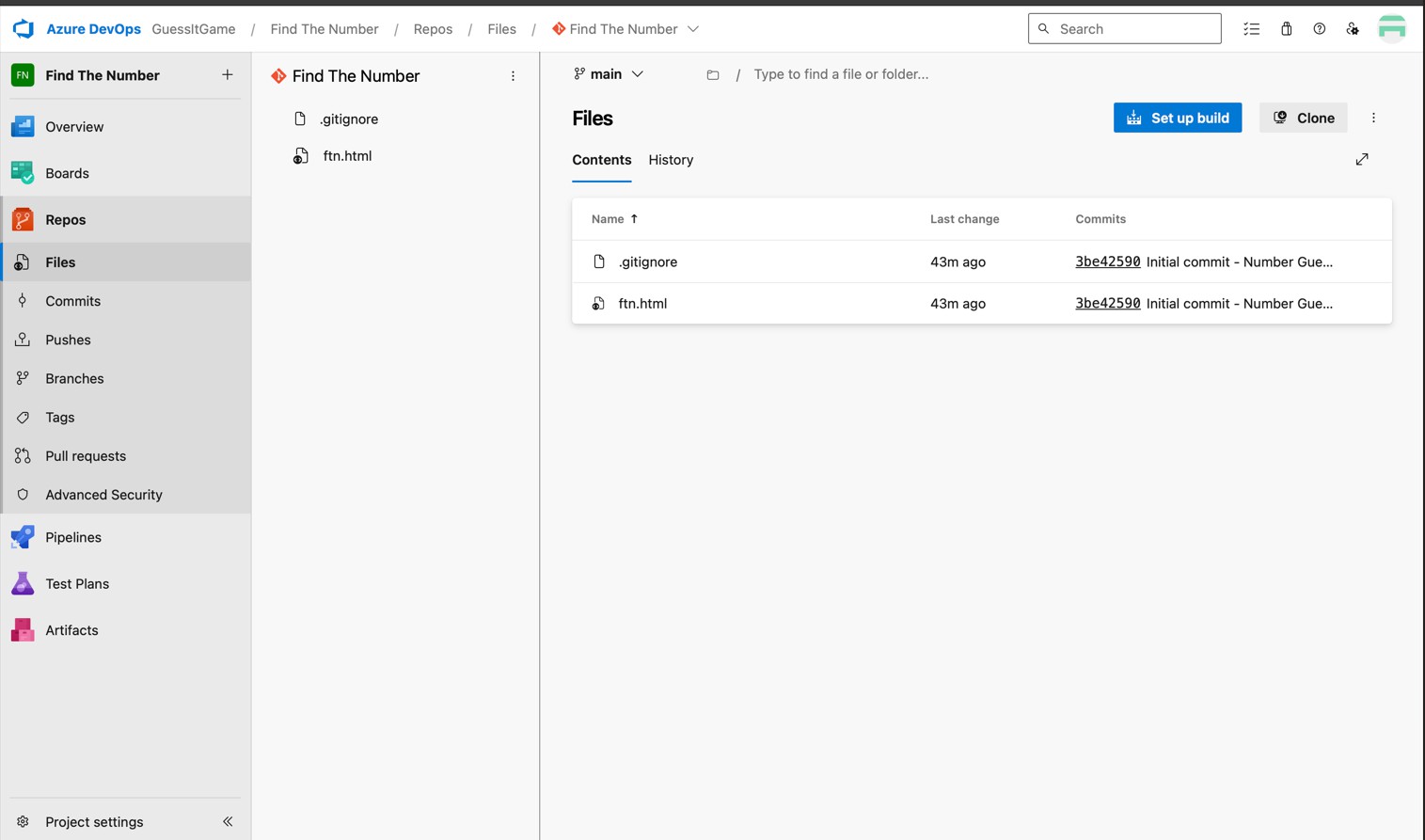
1. Navigating to the repository in Azure DevOps.



1. By clicking the import and pasting the URL to clone the code.



1. After cloning the code to Azure Repos.



## Result :

Thus, a Kanban board was set up, and work-in-progress limits improved task flow.

## Ex.No: 5 Collaborate with teams and stakeholders via chats and notifications.

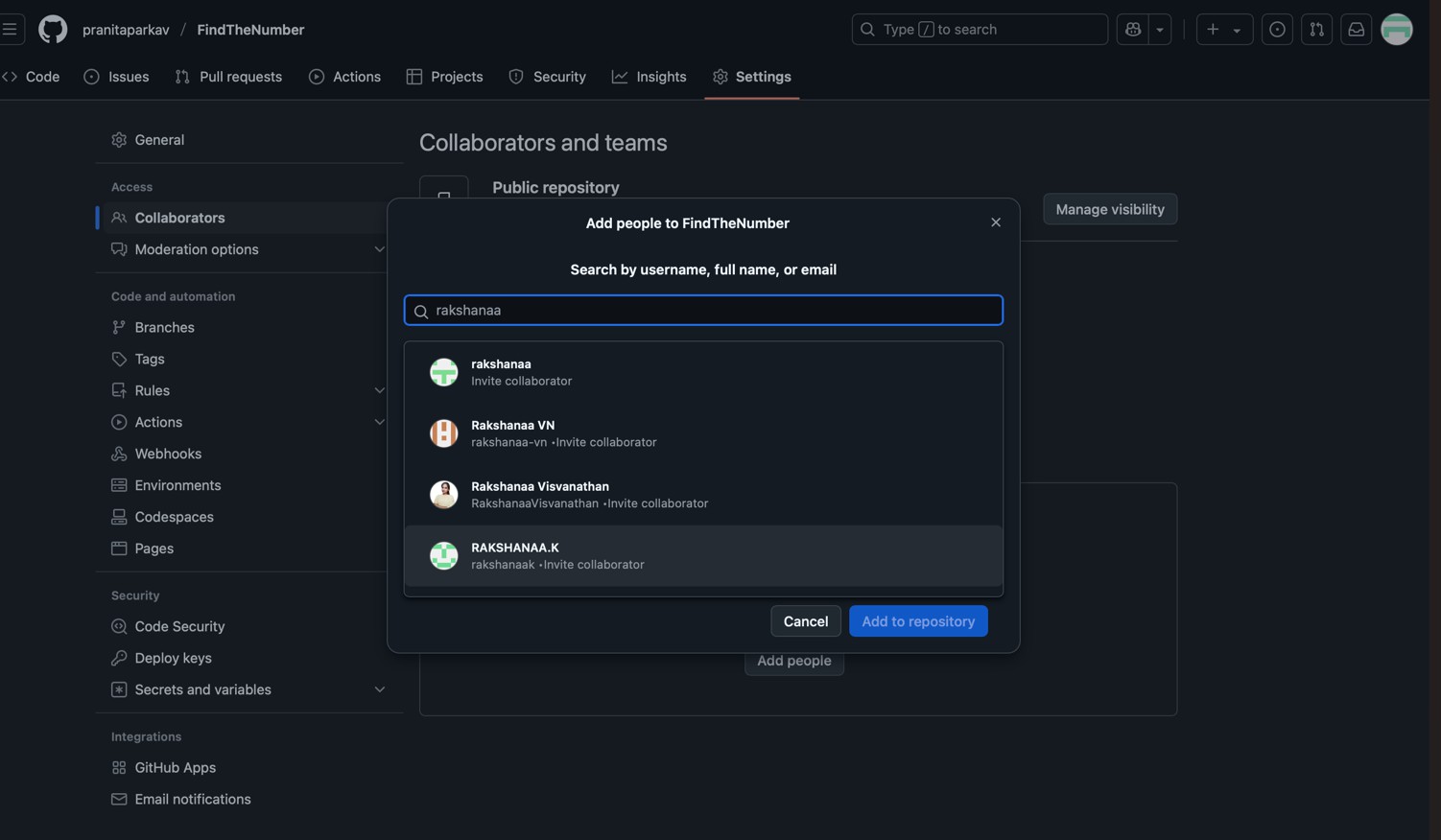
**Date :**

**Aim :**

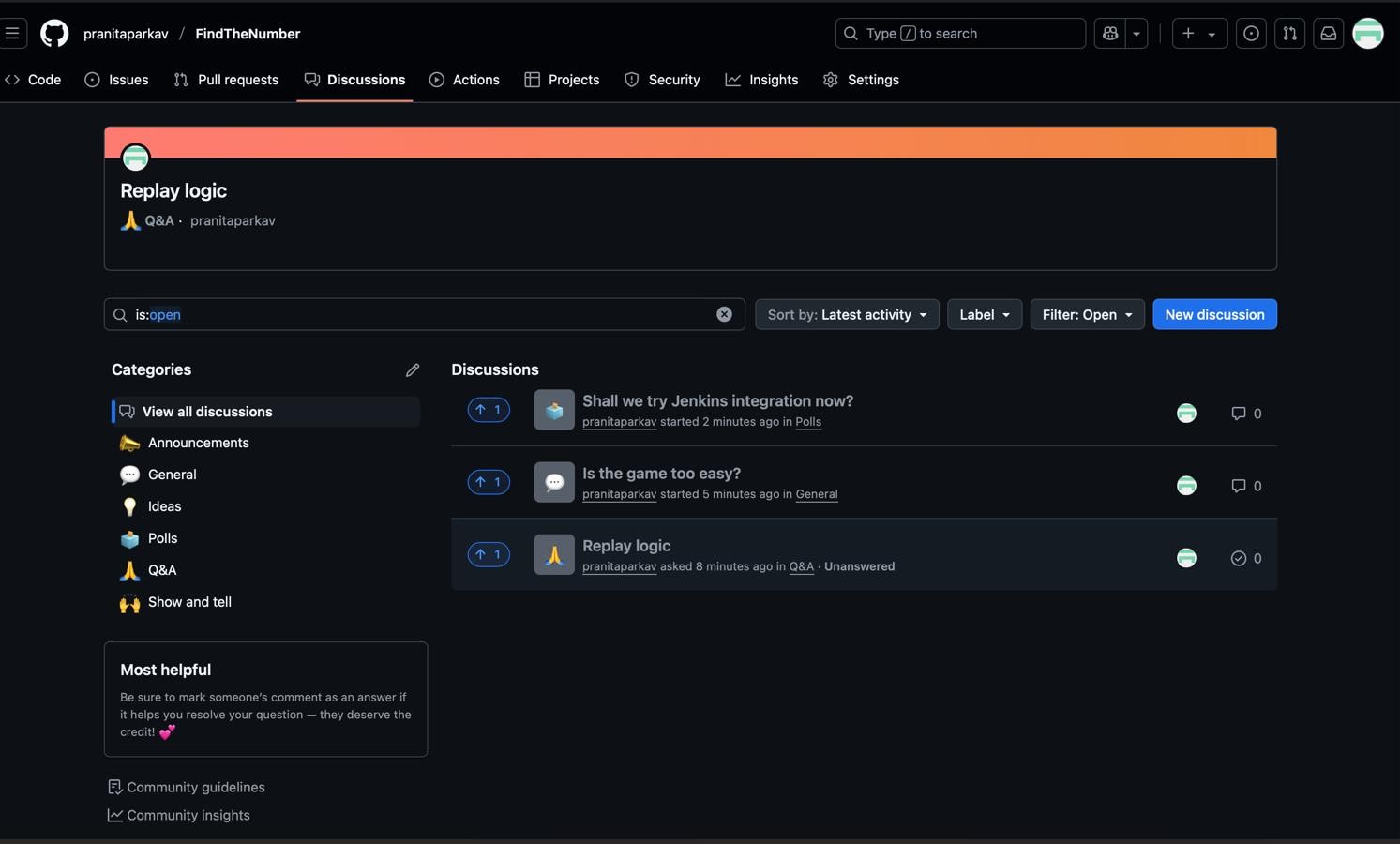
To collaborate with teams and stakeholders through chats and notifications.

**Procedure:**

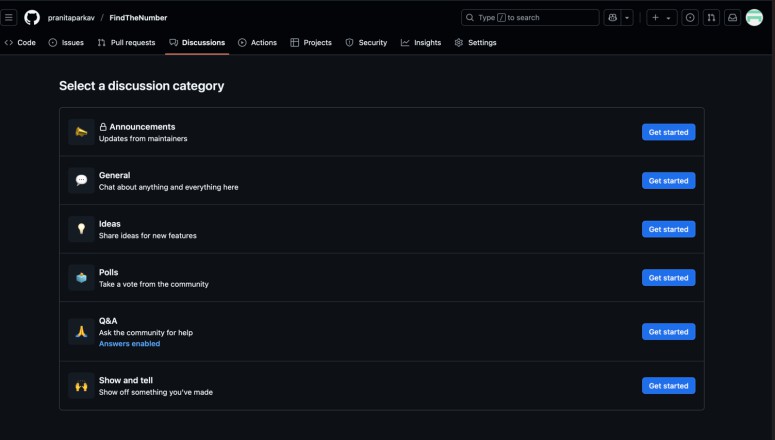
1. Add your team members for collaboration.



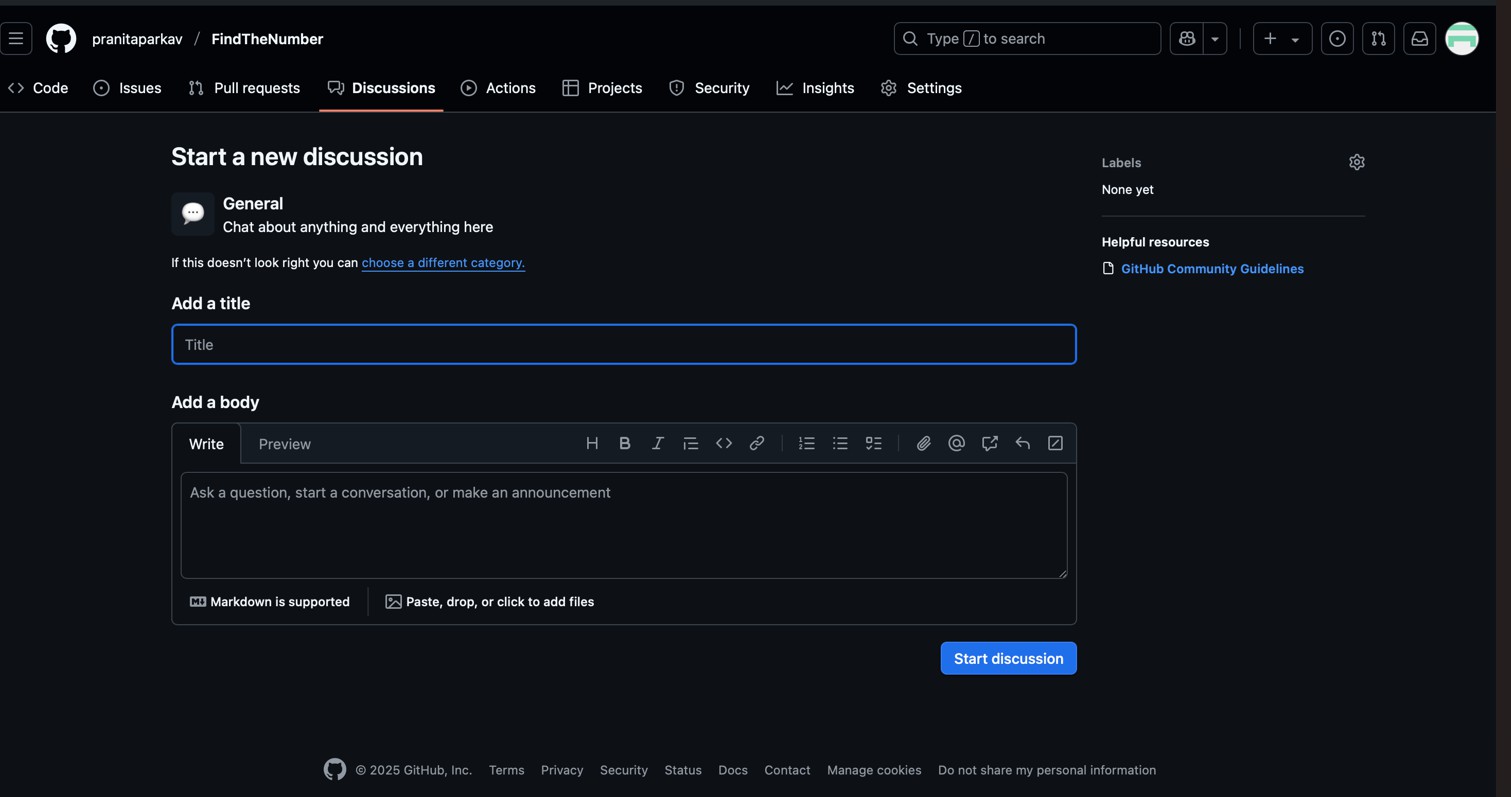
1. Click **Discussions**, then **New discussion**, choose a category, write, and post.



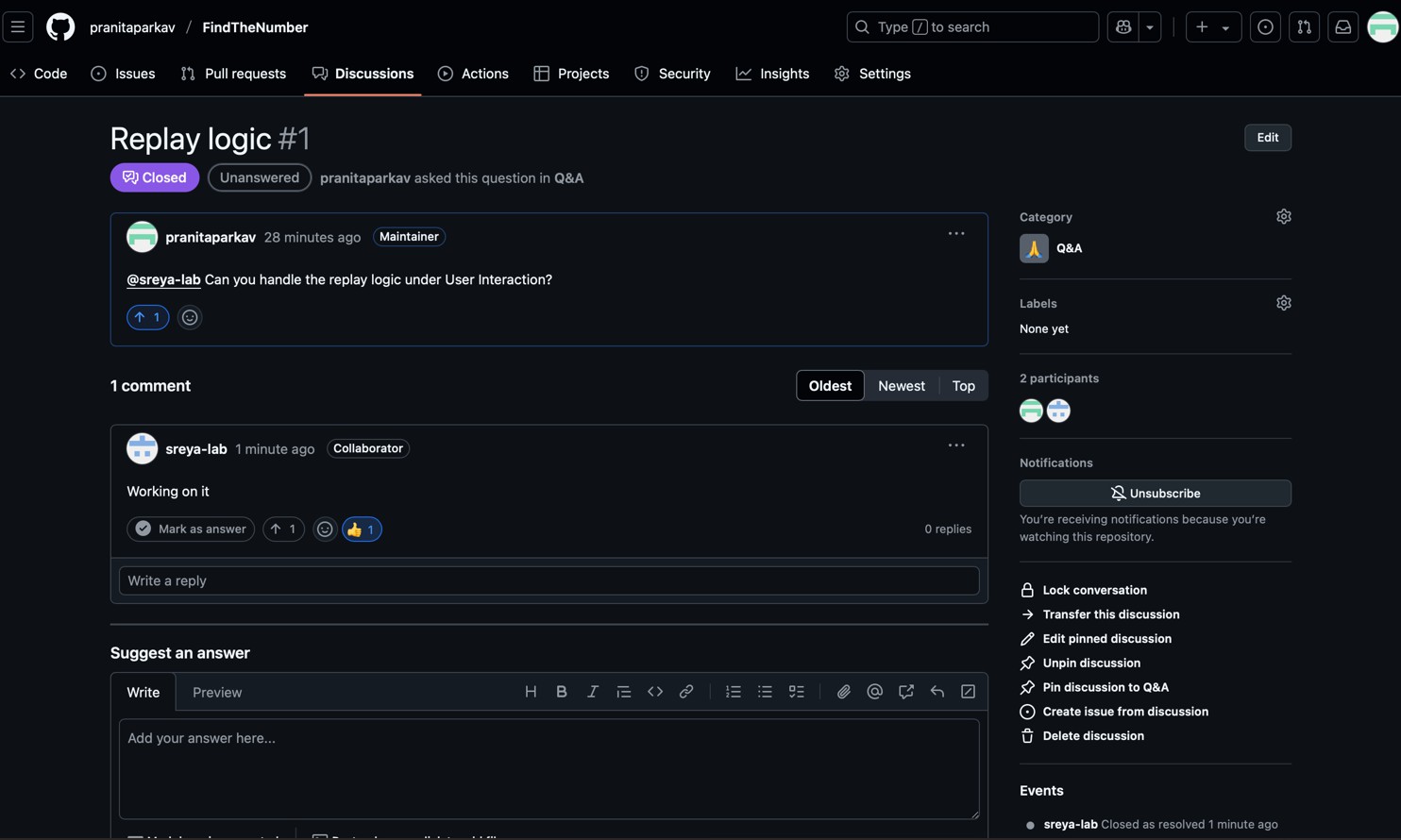
1. Click **Get started** next to a category, then create and post your discussion.



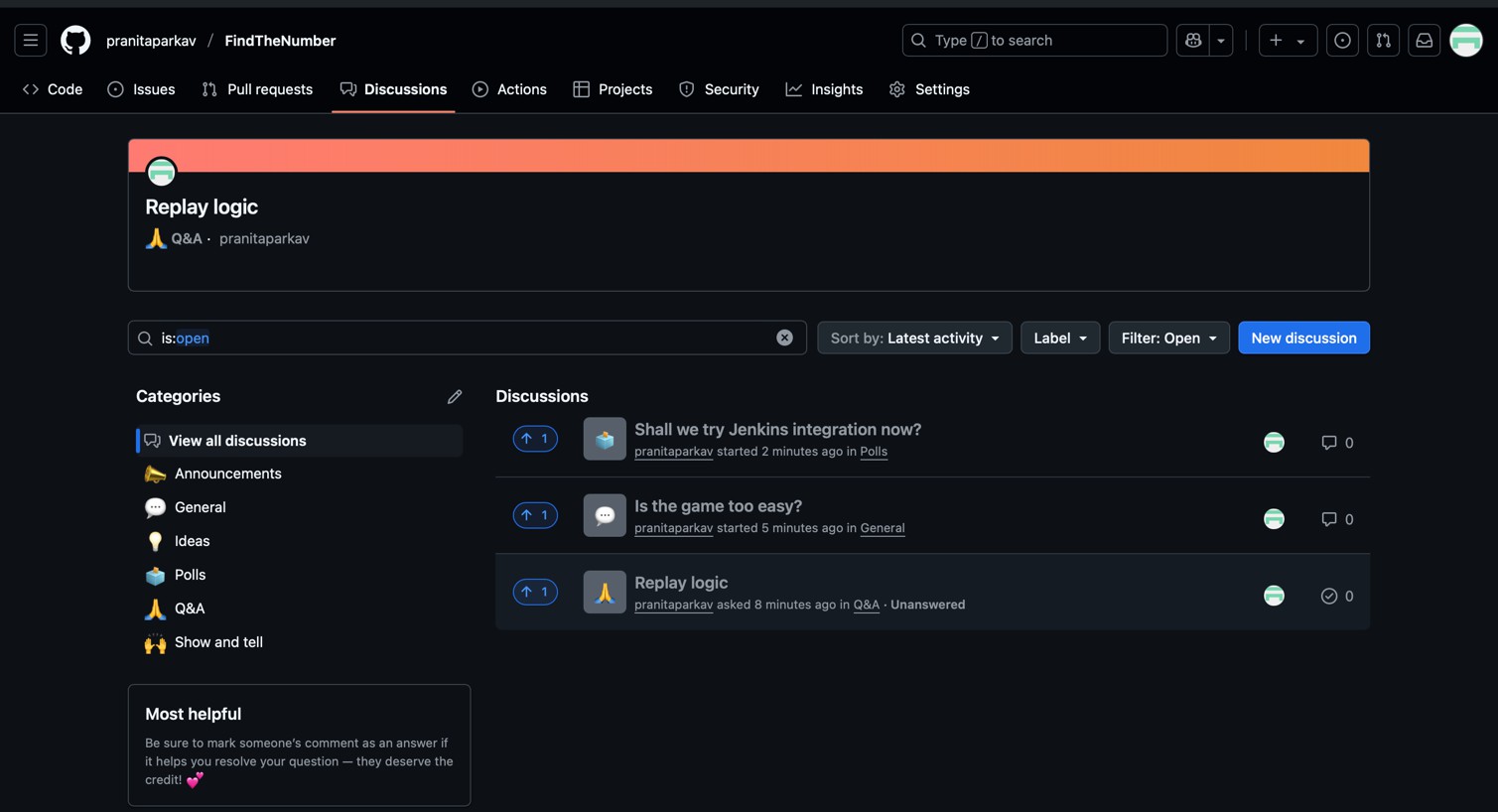
1. Click **Start discussion** to post your issue.



1. Discuss and decide whether to remove the glitching feature or replace its animation with a new one.



1. All discussions viewed; further interaction possible.



## Result :

Thus, teams and stakeholders collaborate through chats and notifications.

## Ex.No: 6 Install and configure Jenkins for continuous integration and delivery.

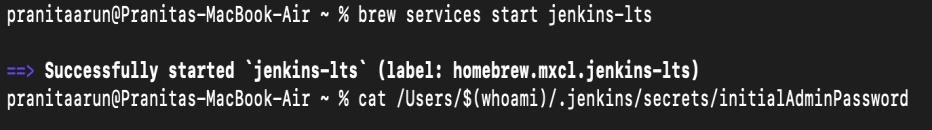
**Date :**

**Aim :**

To install and configure Jenkins for continuous integration and delivery (CI/CD).

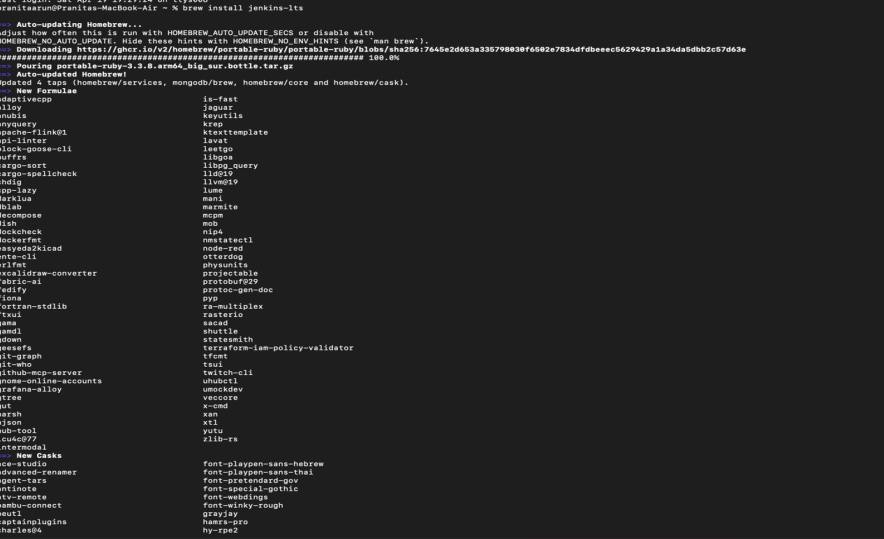
**Procedure:**

1. Download Jenkins.



1. **Command:** java -jar Jenkins.war

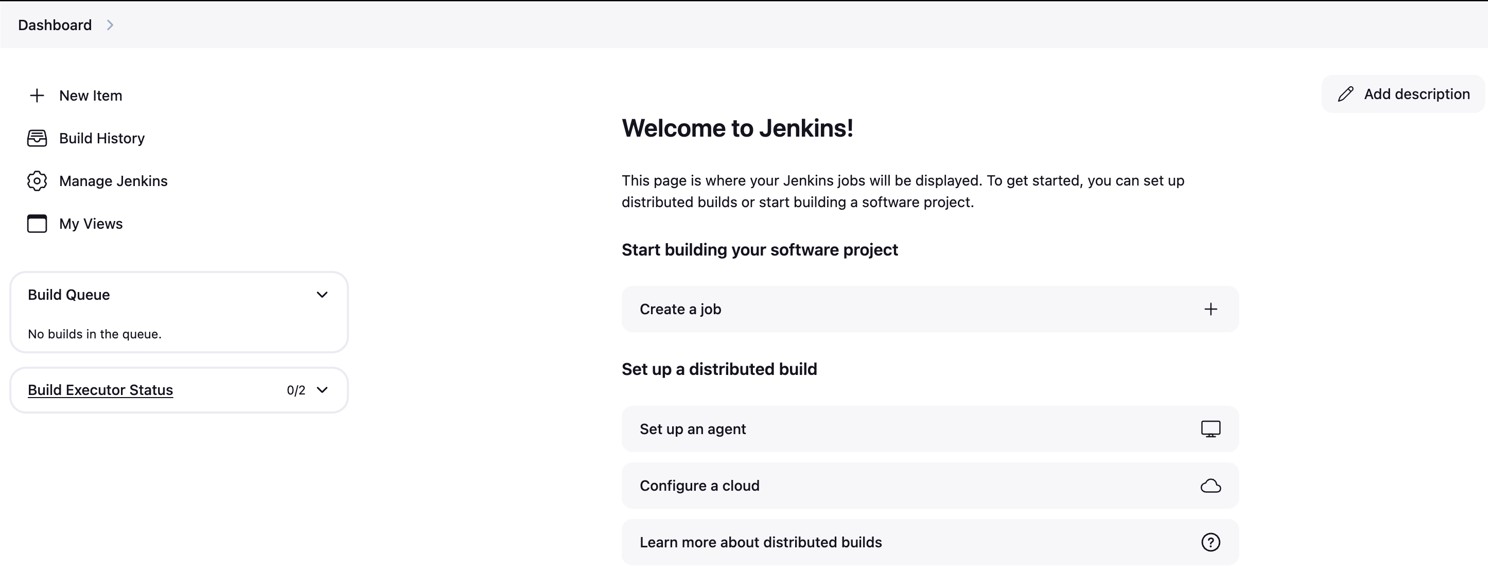
To extract the WAR file.



1. Install the necessary plugins.
2. After installing plugins, create an admin user.



1. After configuration, we can now use CI/CD automation tools.



## Result :

Thus, Jenkins was successfully installed and configured, enabling continuous integration and delivery.

## Ex.No: 7 Set up and configure a continuous integration pipeline.

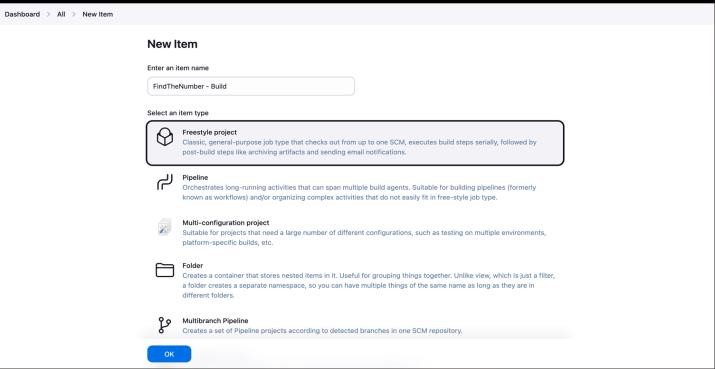
**Date :**

**Aim :**

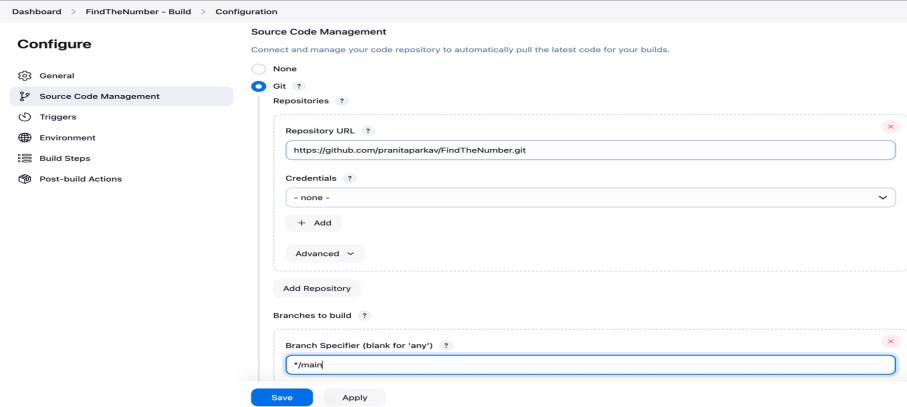
To set up a CI pipeline for automatic code build and testing.

**Procedure:**

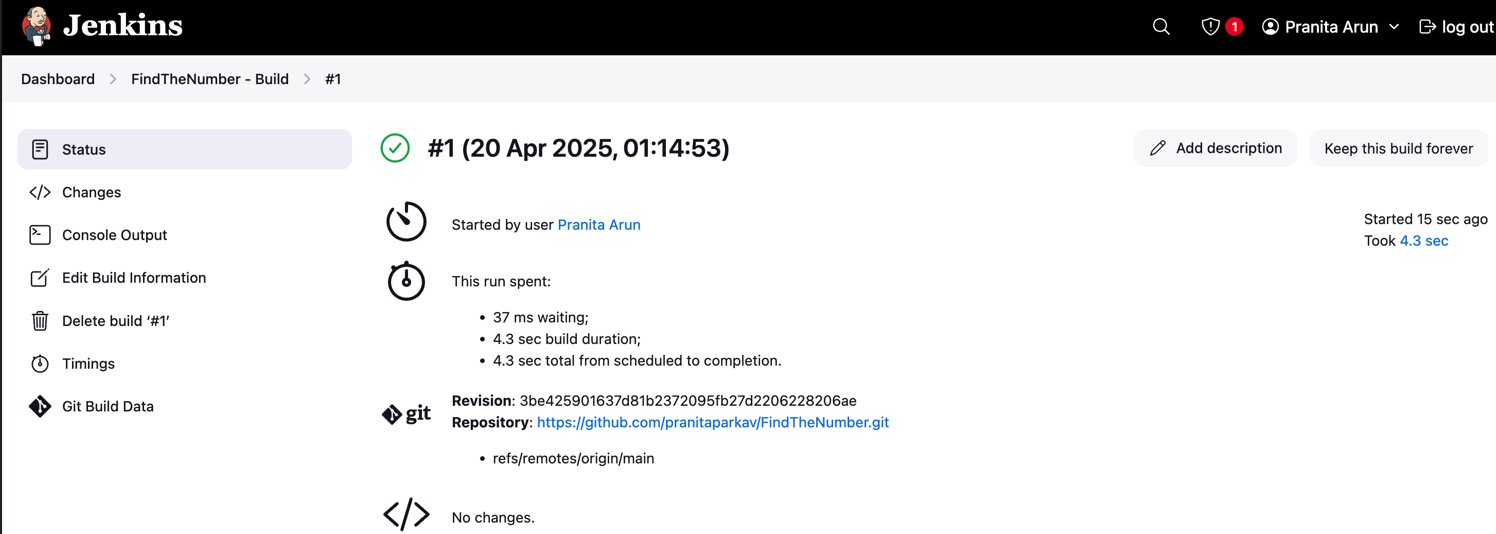
1. To build the project in Jenkins.



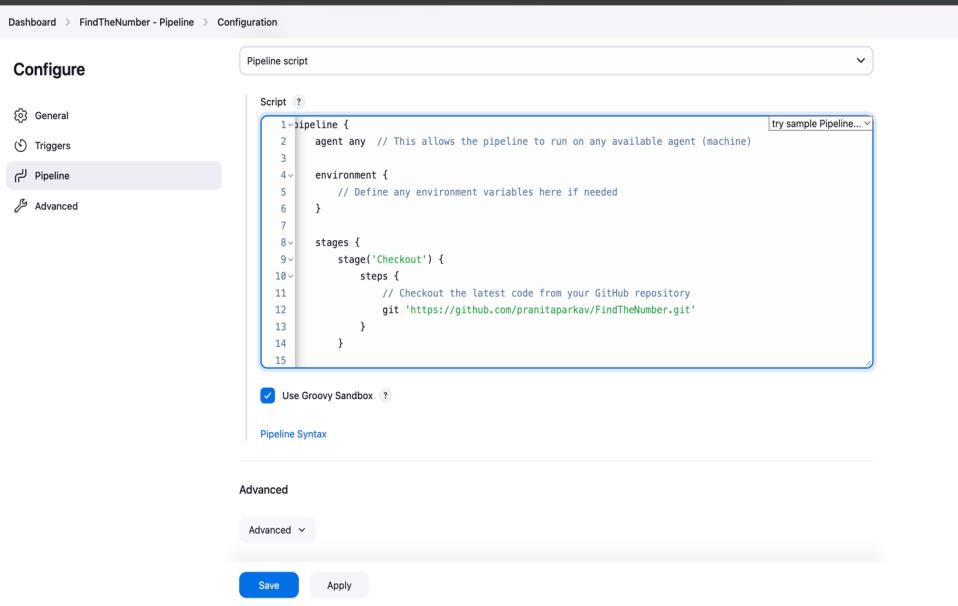
1. In the configuration settings, set the branch to 'main' and add the repository.



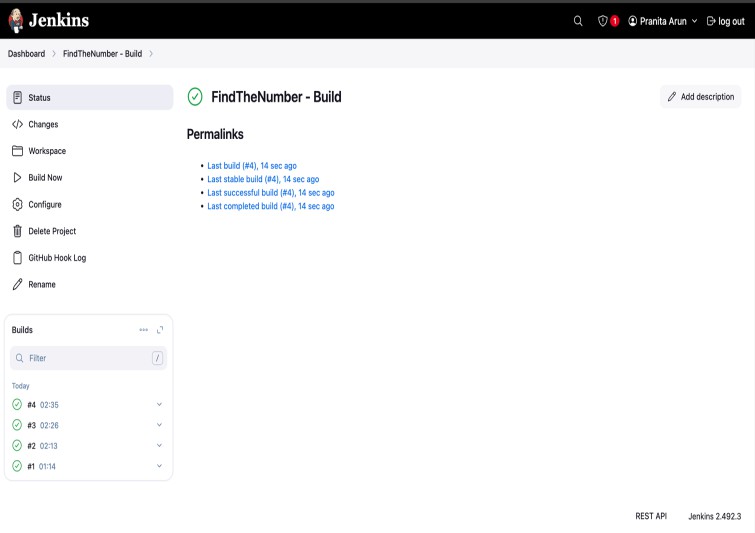
1. After successfully building the repository in Jenkins.



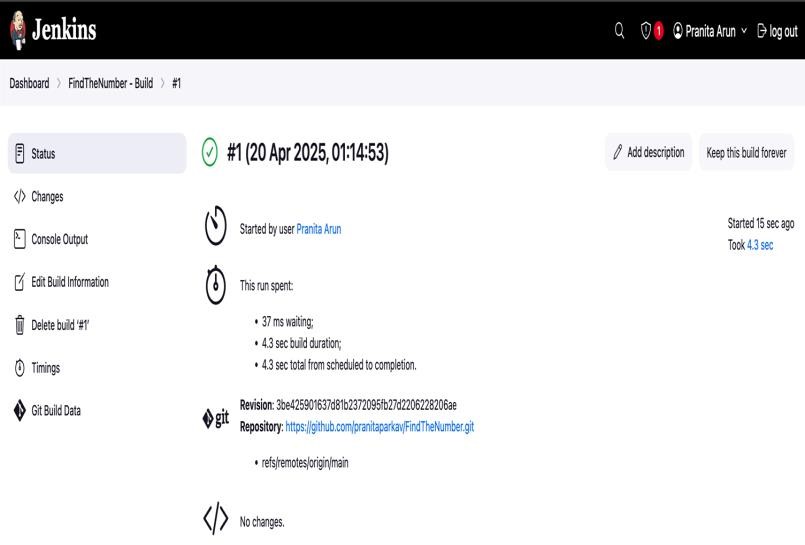
1. Clicking 'New Item' and creating a pipeline.



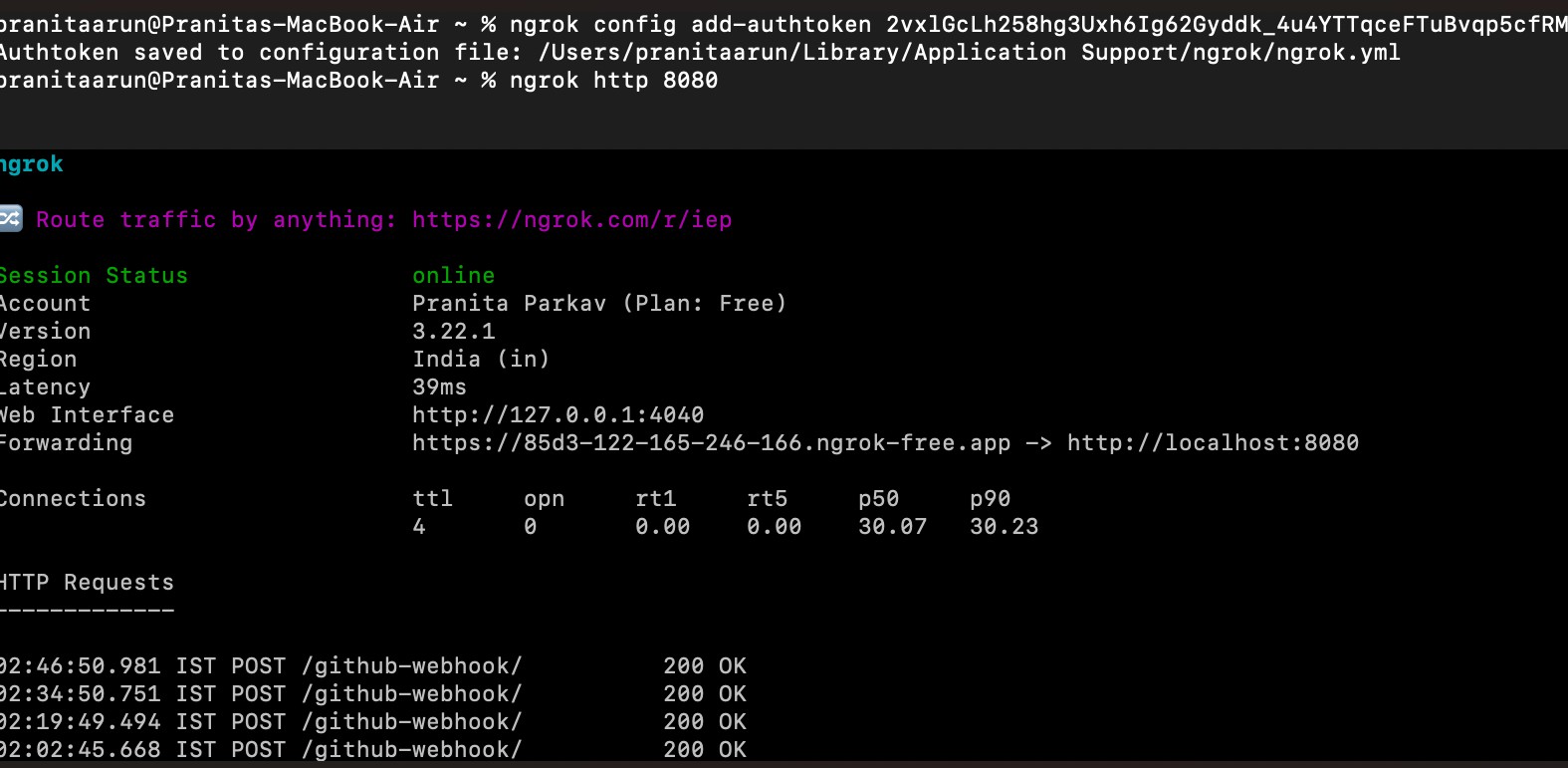
1. After building the pipeline and configuring it.



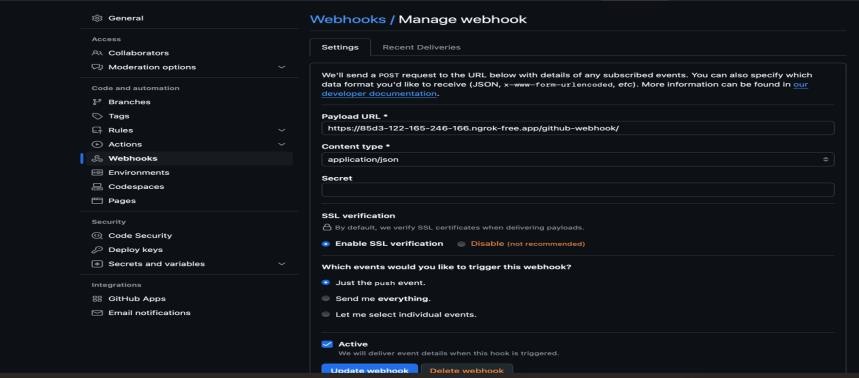
1. Successfully building the pipeline.



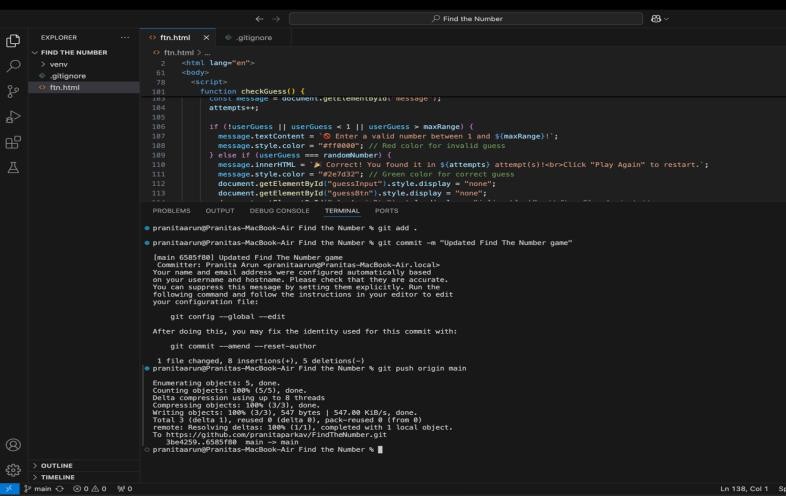
1. Installing Ngrok for automatic triggering.



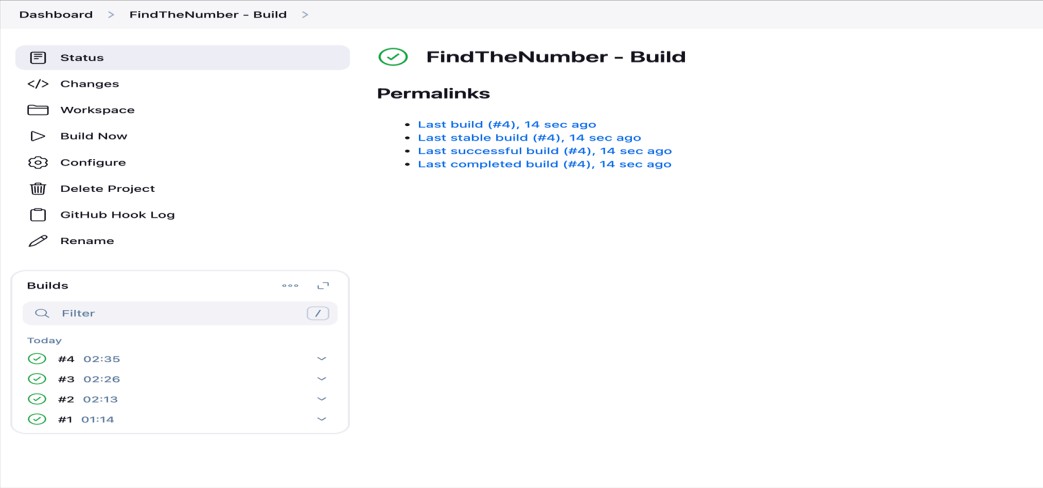
1. Creating the webhook to connect GitHub to Jenkins for automatic triggering.



1. Committing the changes in the code.



1. After committing the changes, it is automatically triggered, and the changes are reflected in Jenkins.



## Result :

The result is automatic building, testing, and deployment of code changes.

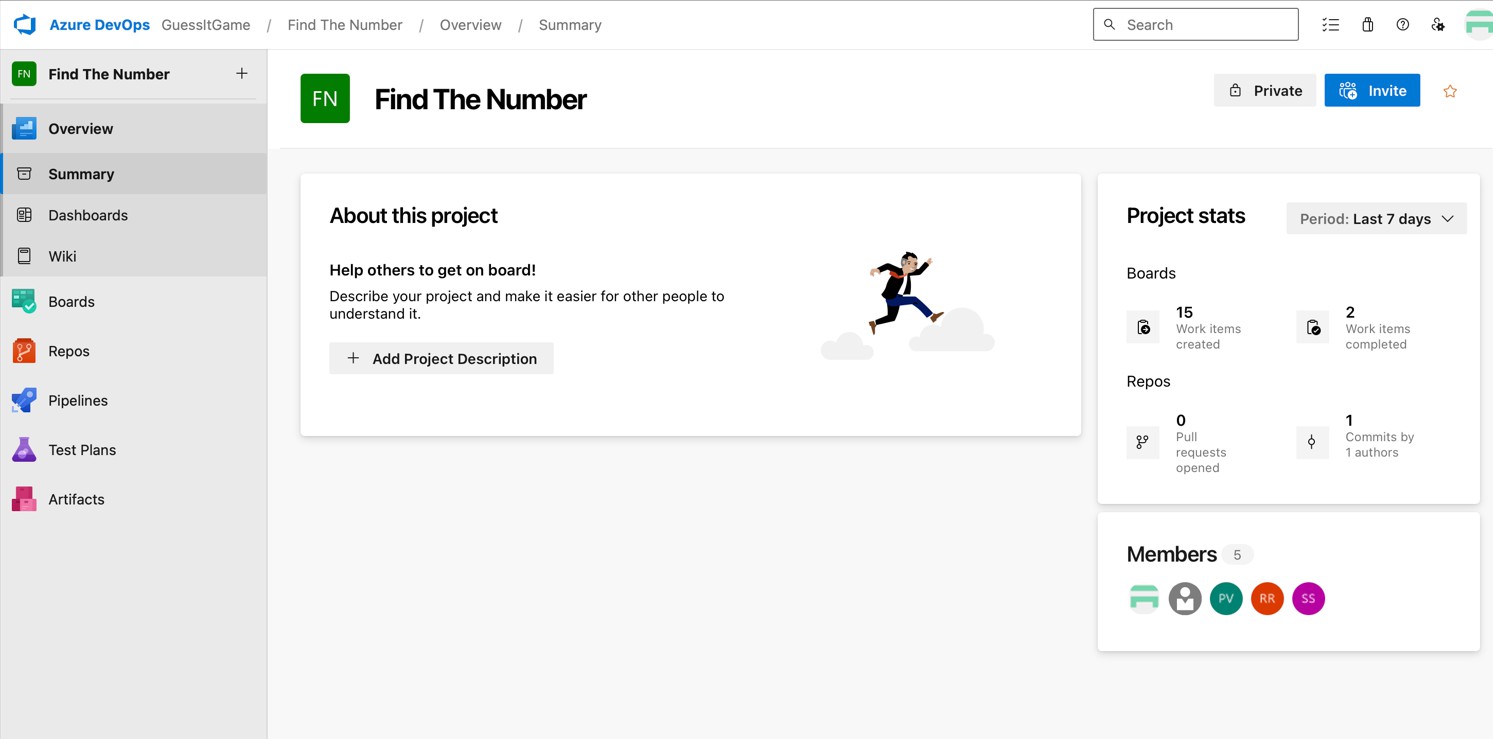
# Ex.No:8 Creating Test Cases in Azure DevOps

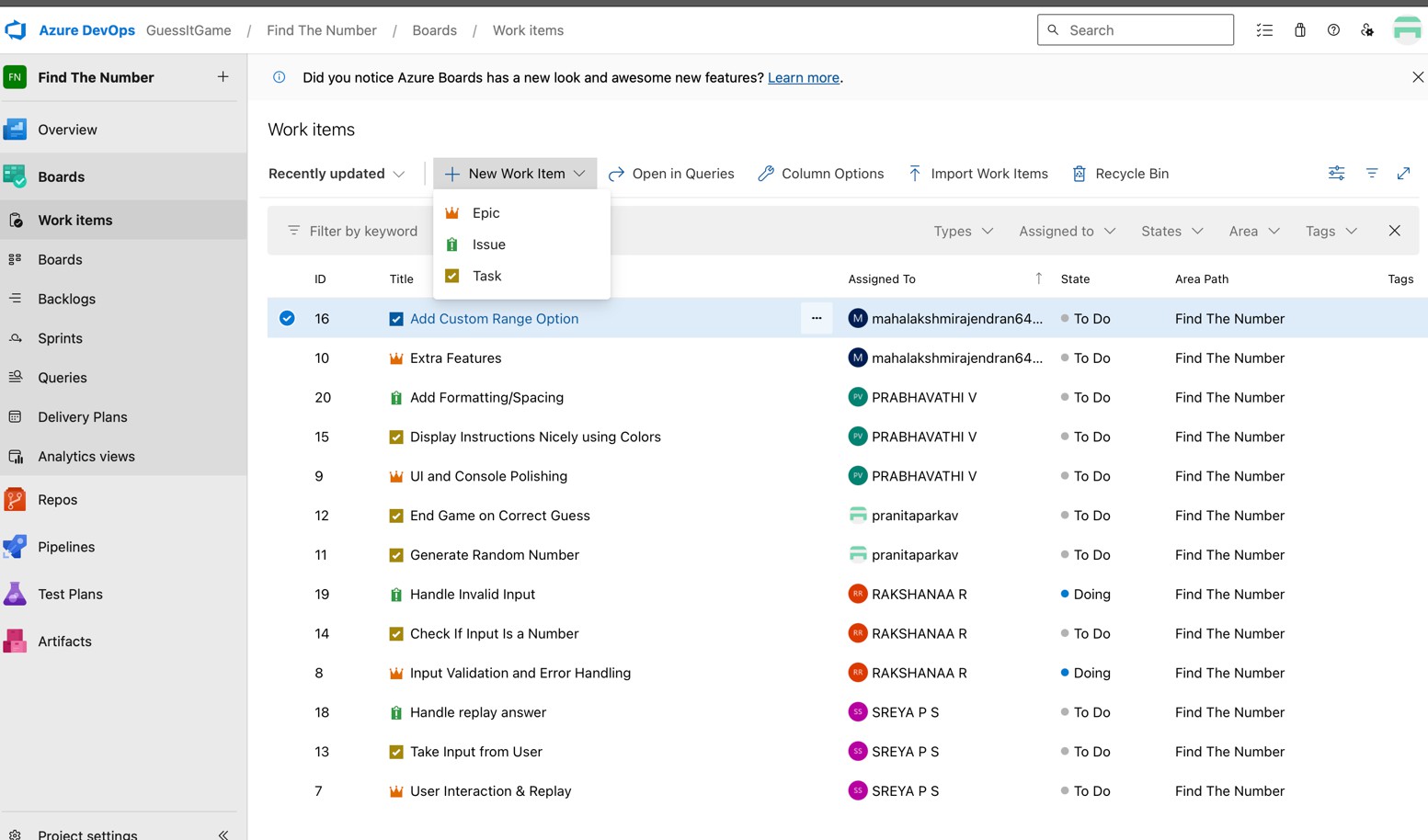
Date:

**Aim:**

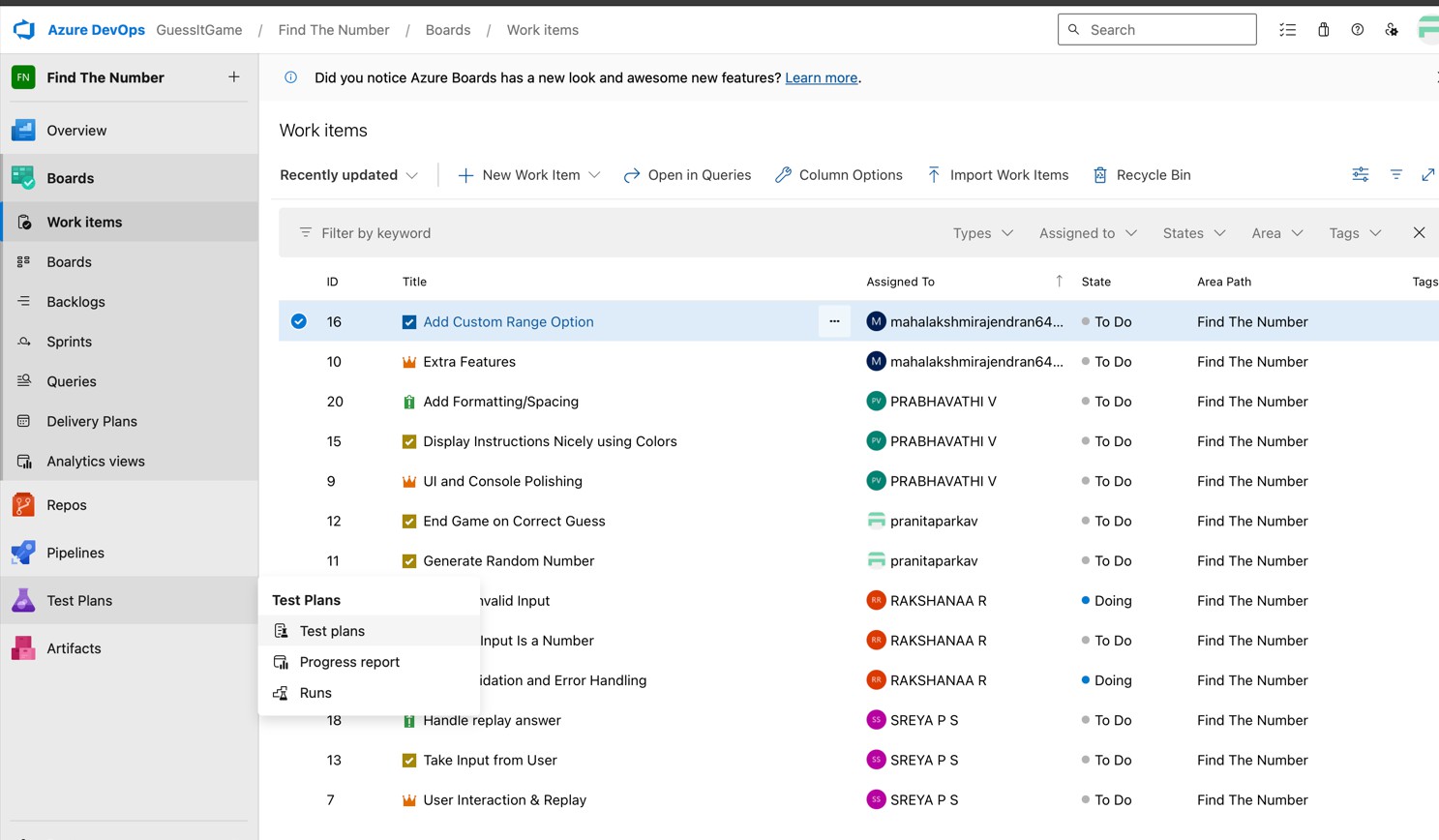
# To create test cases in azure devops.

**Procedure:**

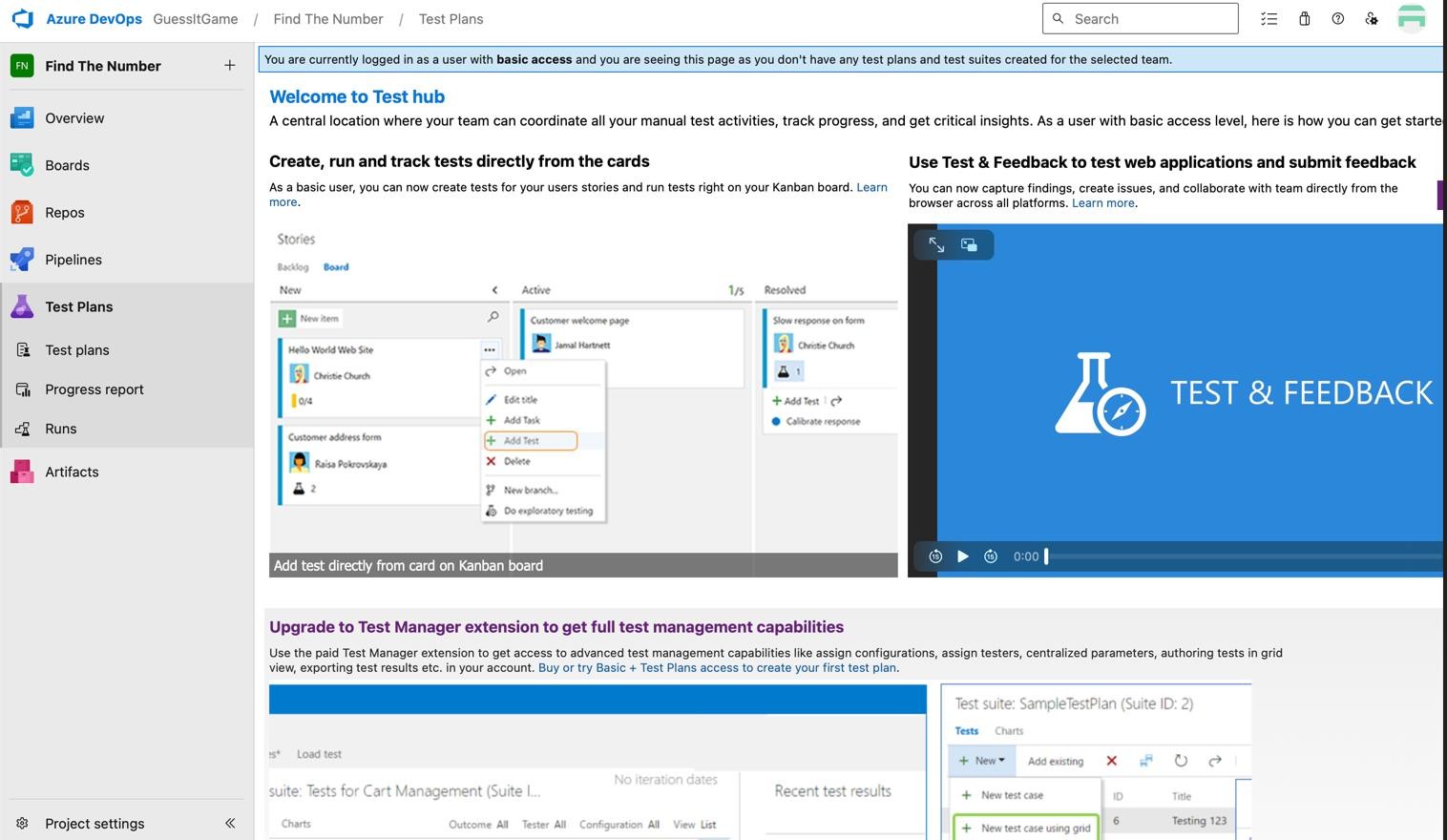
Step 1: Open Your Azure DevOps Project

Step 2: Navigate to Boards → Work Items

Step 3: Enter Test Case Details



Step 4: Save and Repeat for Other Test Cases



**Result:**

# Thus the test cases in azure devops have been created successfully.

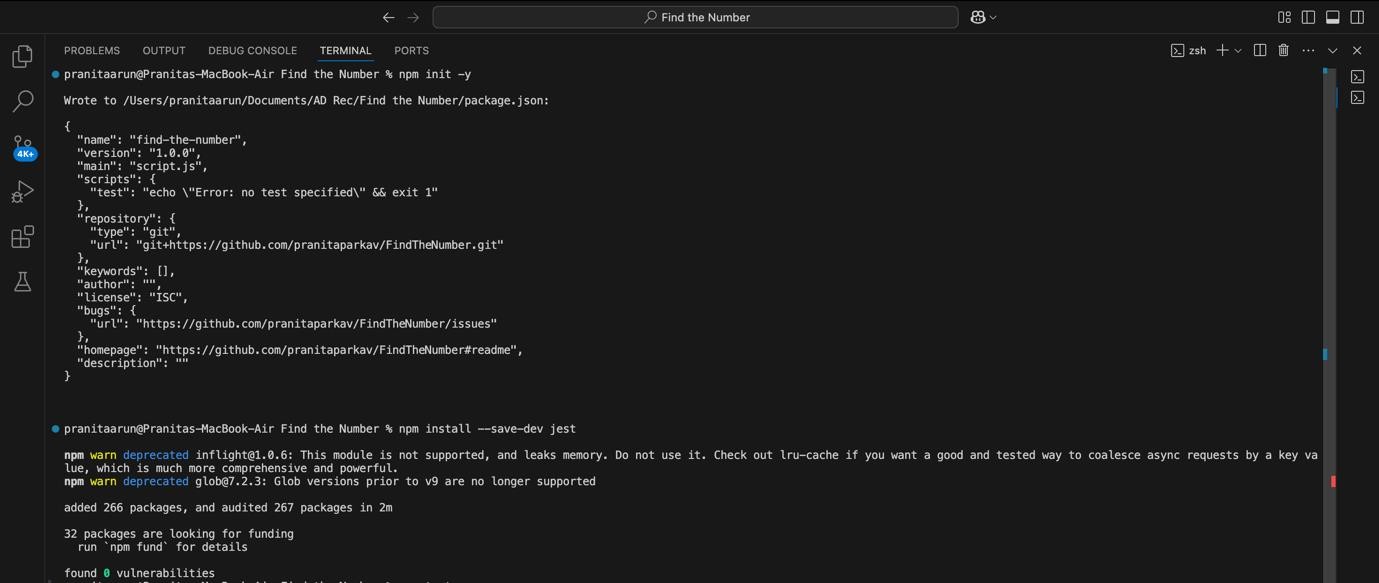
**Ex.No: 9 Implement Continuous Testing and Feedback to Identify and Address Issues Date :**

**Aim :**

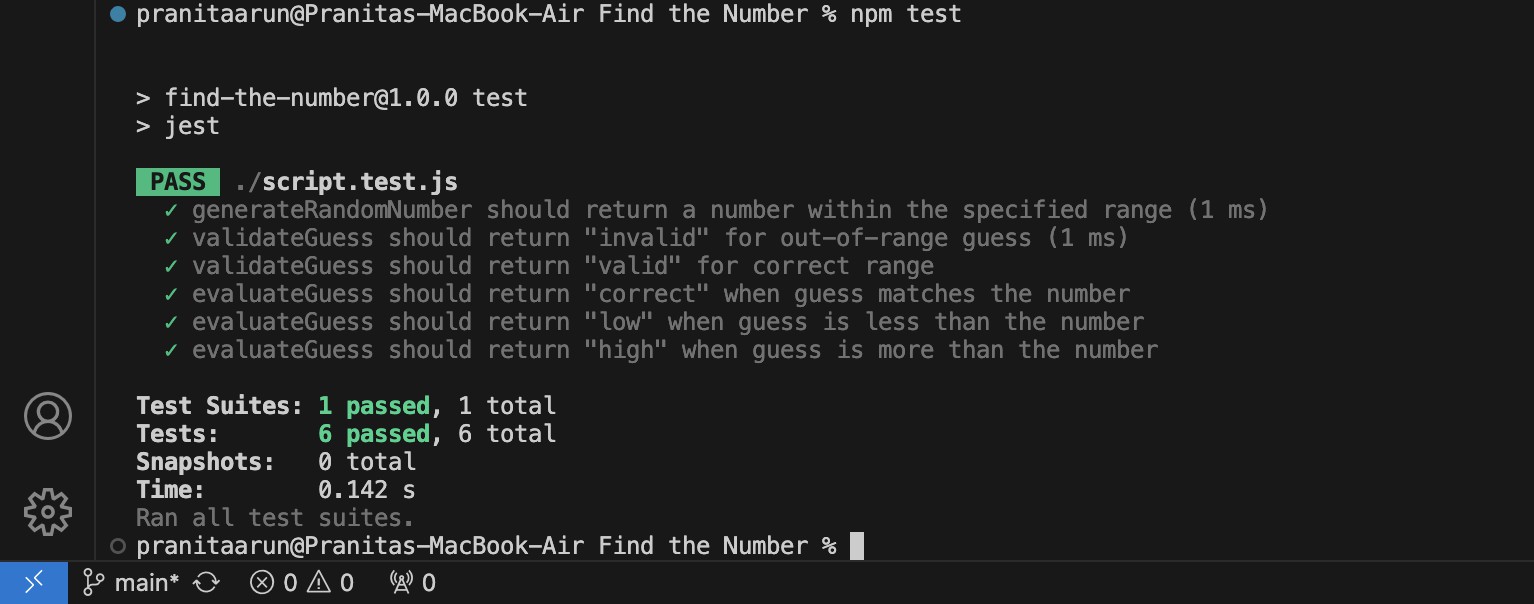
To implement continuous testing and feedback in the CI/CD pipeline to identify and address issues early in the development process.

**Procedure:**

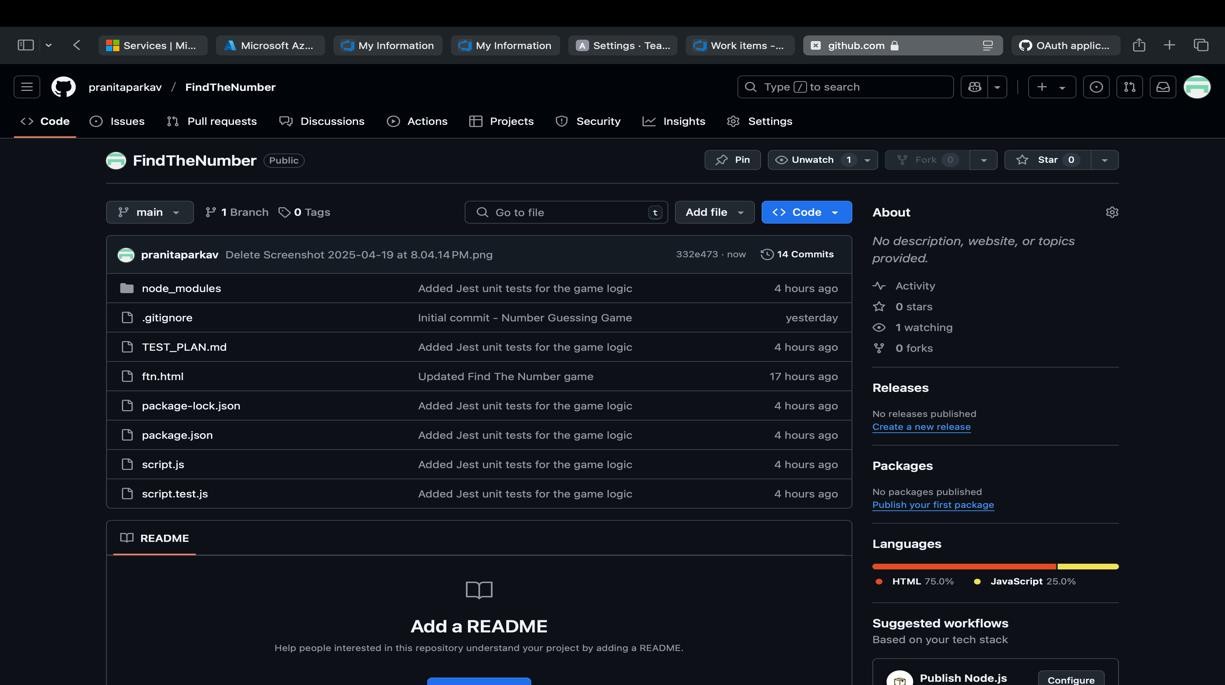
1. Open terminal in your project folder and run to create package.json , node\_modules/ , package-lock.json



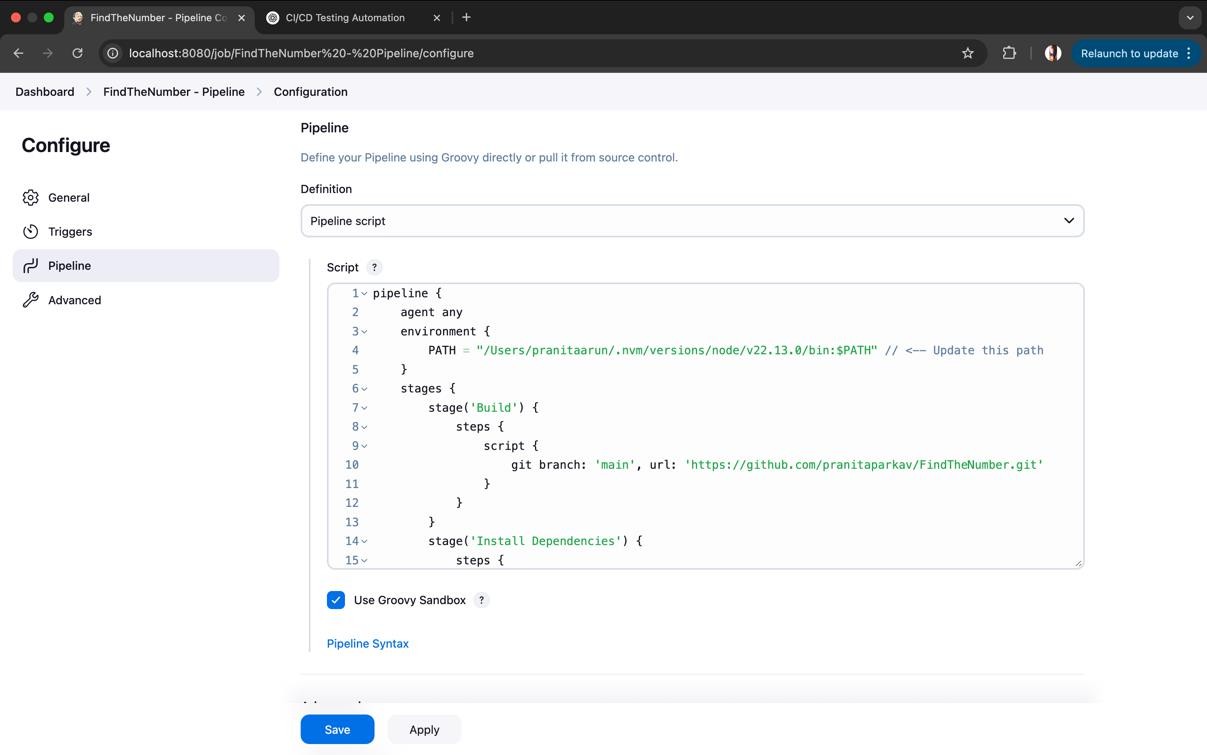
1. Create a script.test.js and test locally.



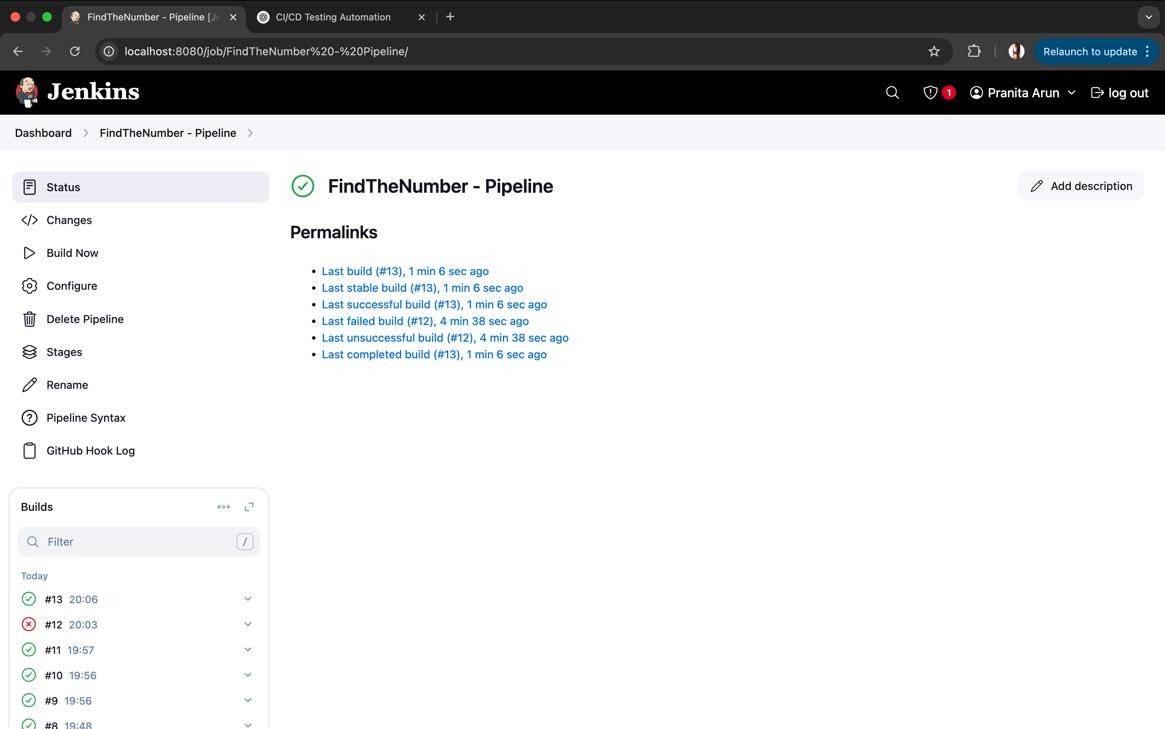
1. Push the newly created files into the GitHub repo.



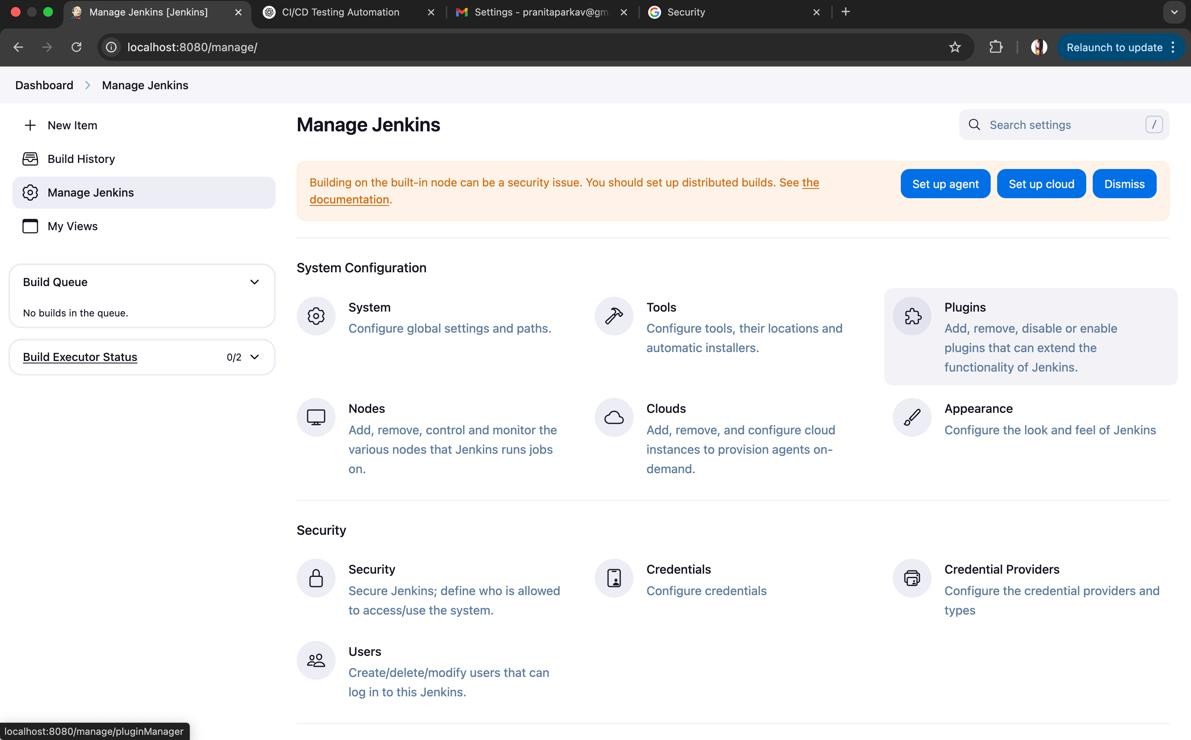
1. Add npm path to Jenkins' environment – pipeline script.



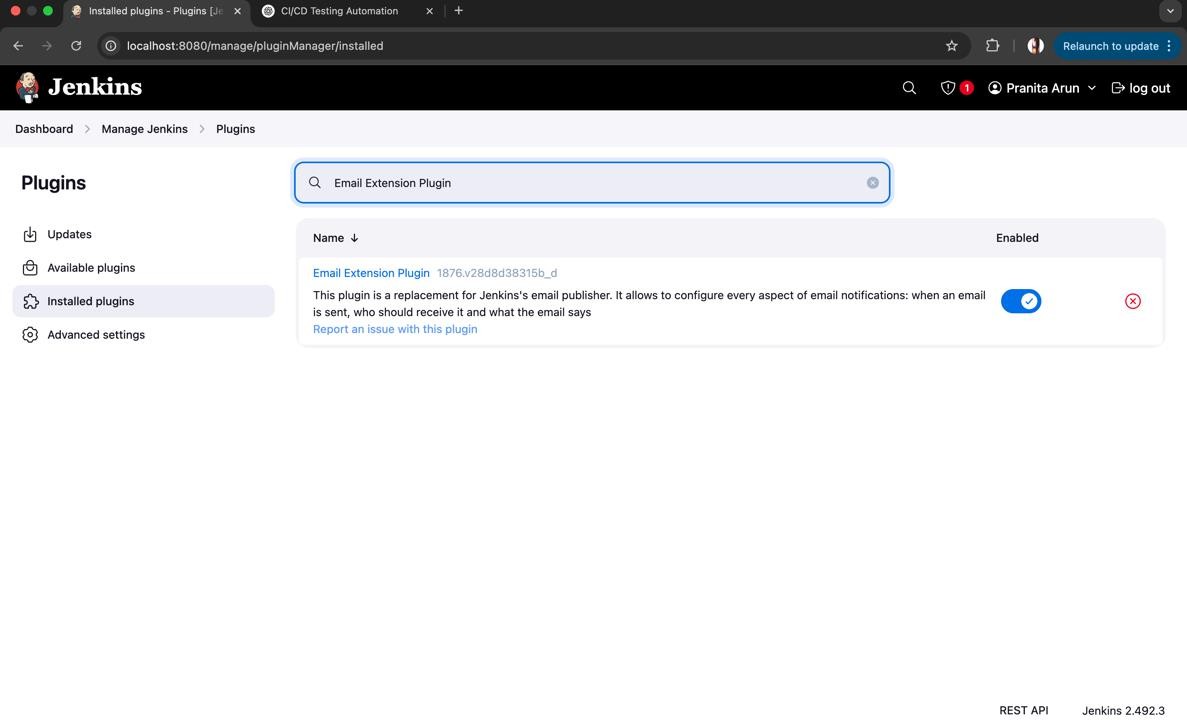
1. Save and build now.



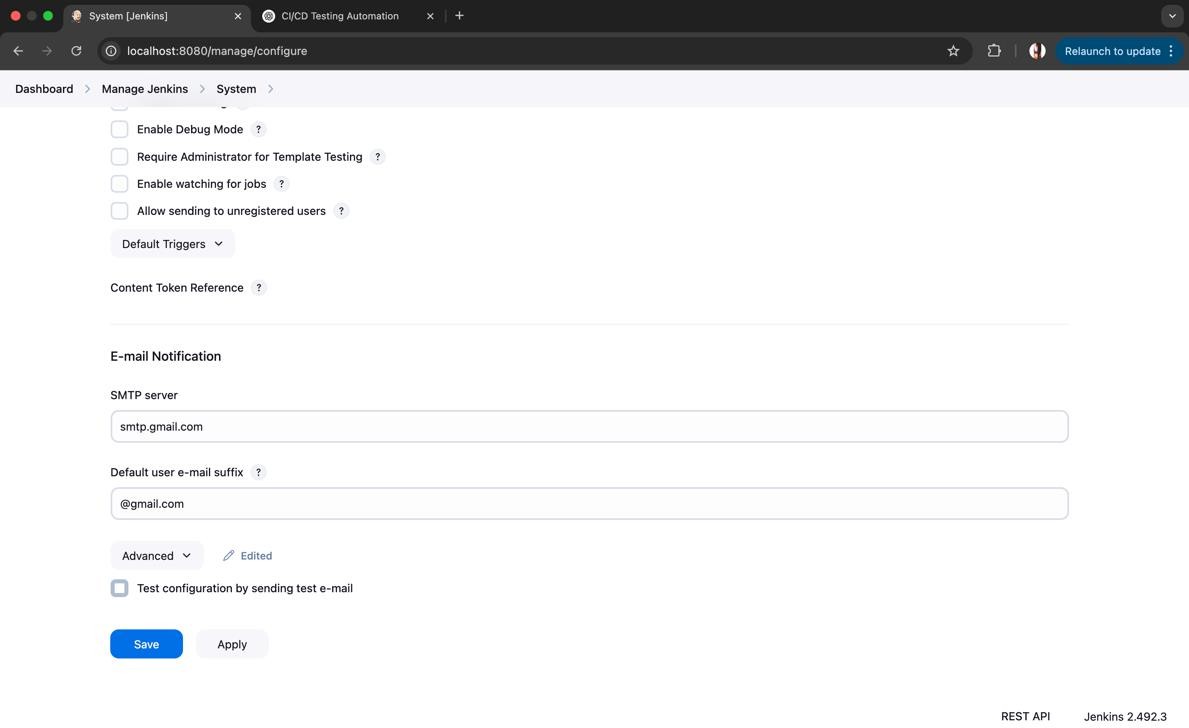
1. Go to Manage Jenkins and select Plugins.



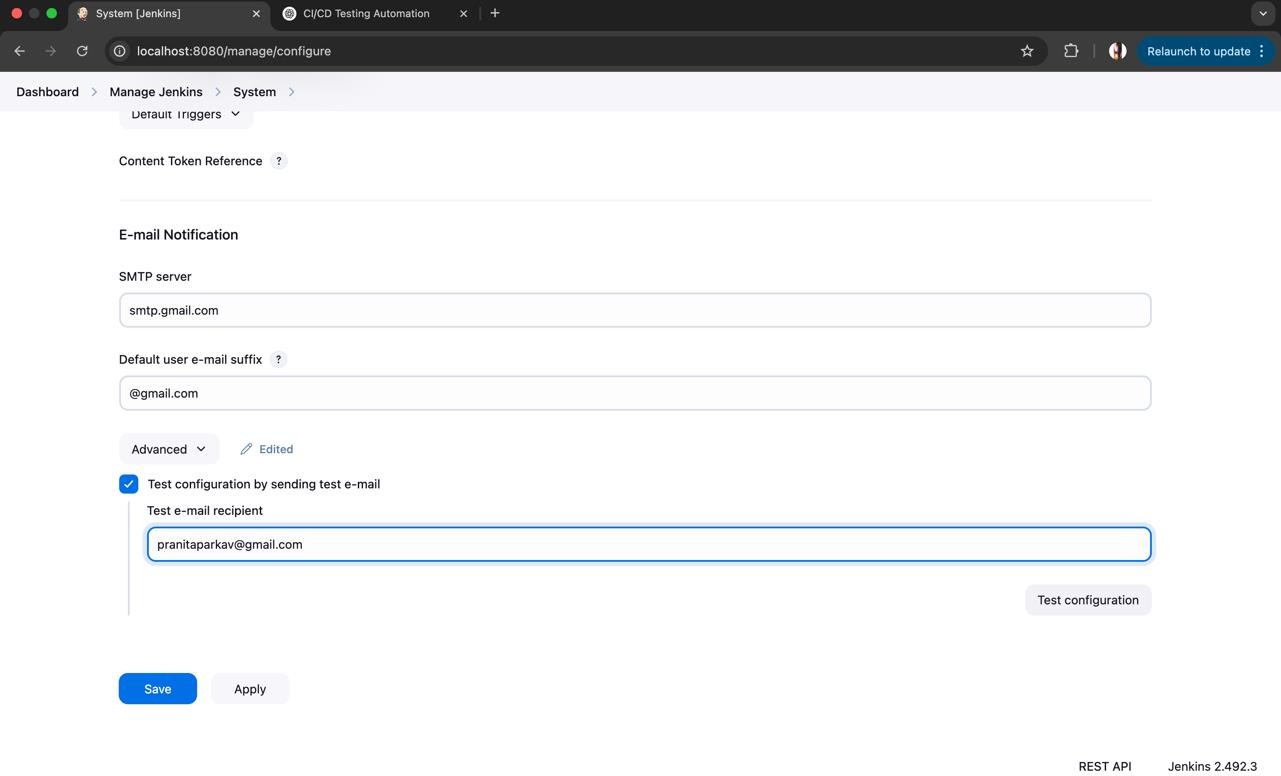
1. Make sure Email Extension Plugin is enabled.



1. Fill the email section with appropriate details.



1. Test configuration by sending test e-mail.



RESULT

Thus, continuous testing and feedback in the CI/CD pipeline to identify and address issues early in the development process has been implemented.

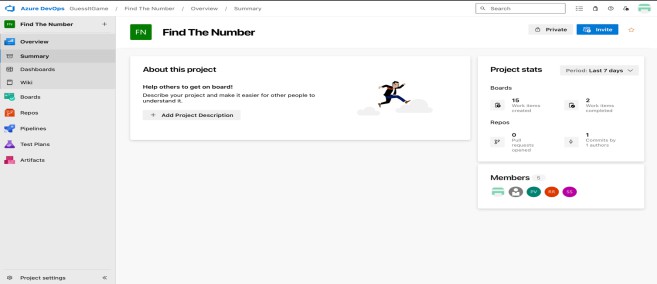
## Ex.No: 10 Set up and use monitoring dashboards to track project performance and health.

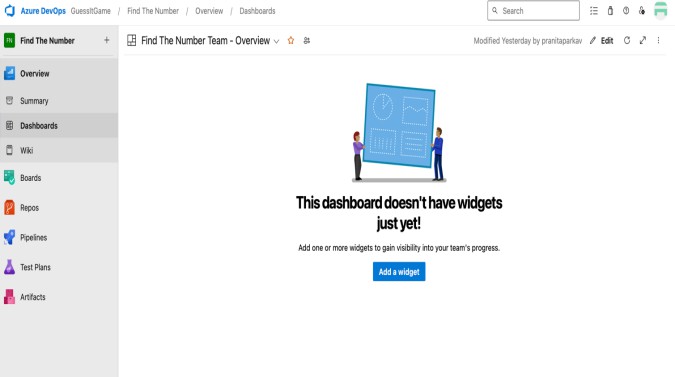
**Date :**

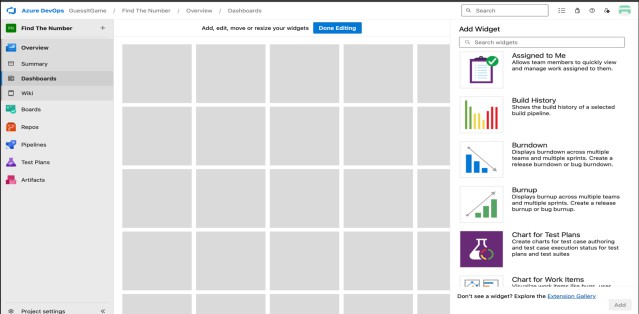
**Aim :**

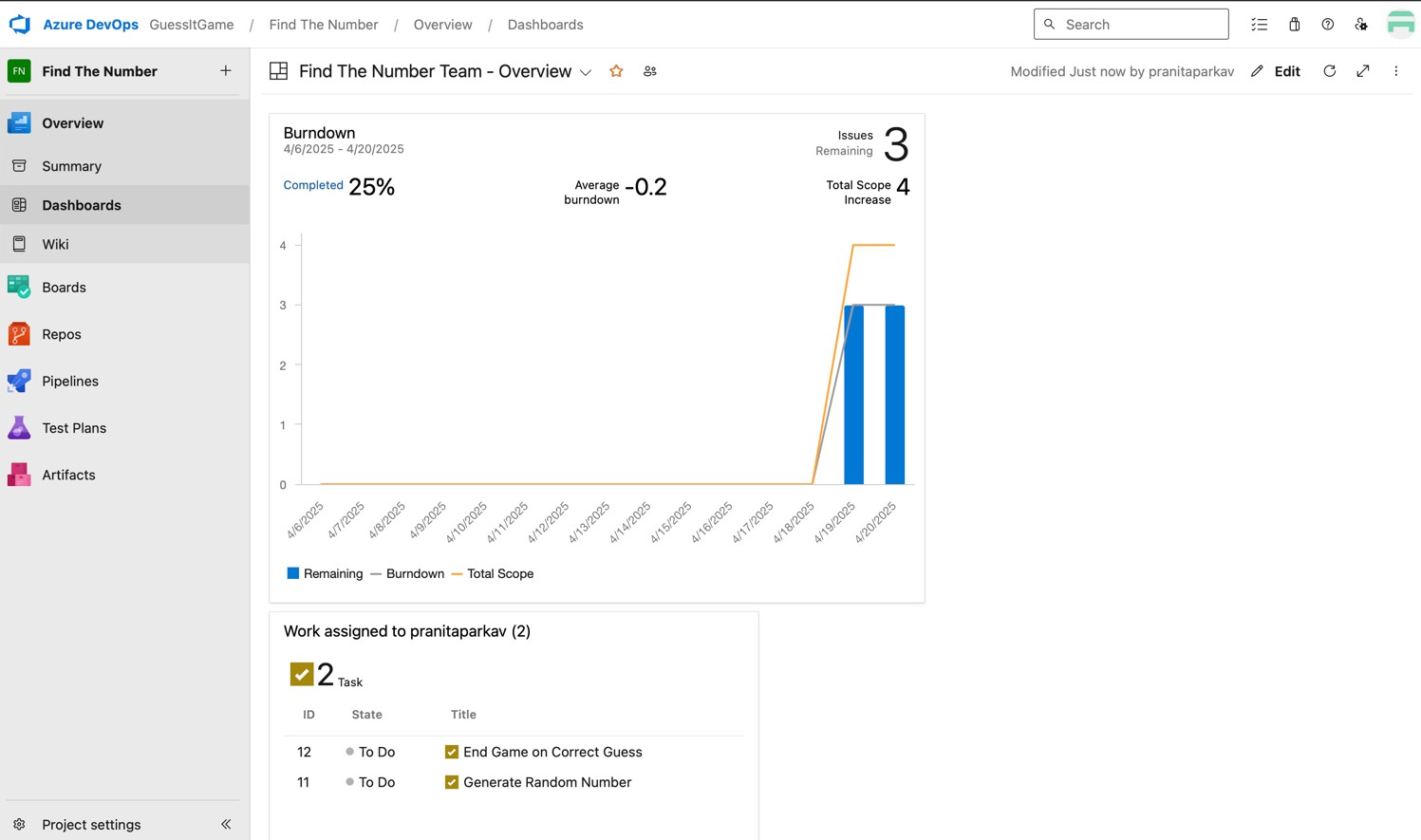
To track project performance and health using monitoring dashboards.

**Procedure:**

1. Open your Azure DevOps project.
2. Go to the Dashboards section from the left menu.



1. Click “+” to create or open an existing dashboard.
2. Configure each widget with the relevant team, sprint, and queries.
3. Use charts like Burndown, Burnup, and CFD to monitor progress.



**8.** Save the dashboard and share it with your team.

## Result :

Thus, monitoring dashboards were set up and used to effectively track project performance and health.