INDEX

|  |  |  |
| --- | --- | --- |
| **S.NO** | **DATE** | **TITLE** |
| **1.** | **21/1/25** | **AzureDevOpsEnvironmentSetup** |
| **2.** | **21/1/25** | **AzureDevOpsProjectSetupandUserStoryManagement** |
| **3.** | **28/1/25** | **SettingUpEpics,FeaturesandUserStoriesforProjectPlanning** |
| **4.** | **11/2/25** | **SprintPlanning** |
| **5.** | **18/2/25** | **PokerEstimation** |
| **6.** | **25/2/25** | **DesigningClassandSequenceDiagramsforProjectArchitecture** |
| **7.** | **04/3/25** | **DesigningUse-CaseandActivityDiagramsforProjectArchitecture** |
| **8.** | **25/3/25** | **Testing–TestPlansandTestCases** |
| **9.** | **15/4/25** | **CI/CDPipelinesinAzure** |
| **10.** | **22/4/25** | **GitHub:ProjectStructure&NamingConvention** |

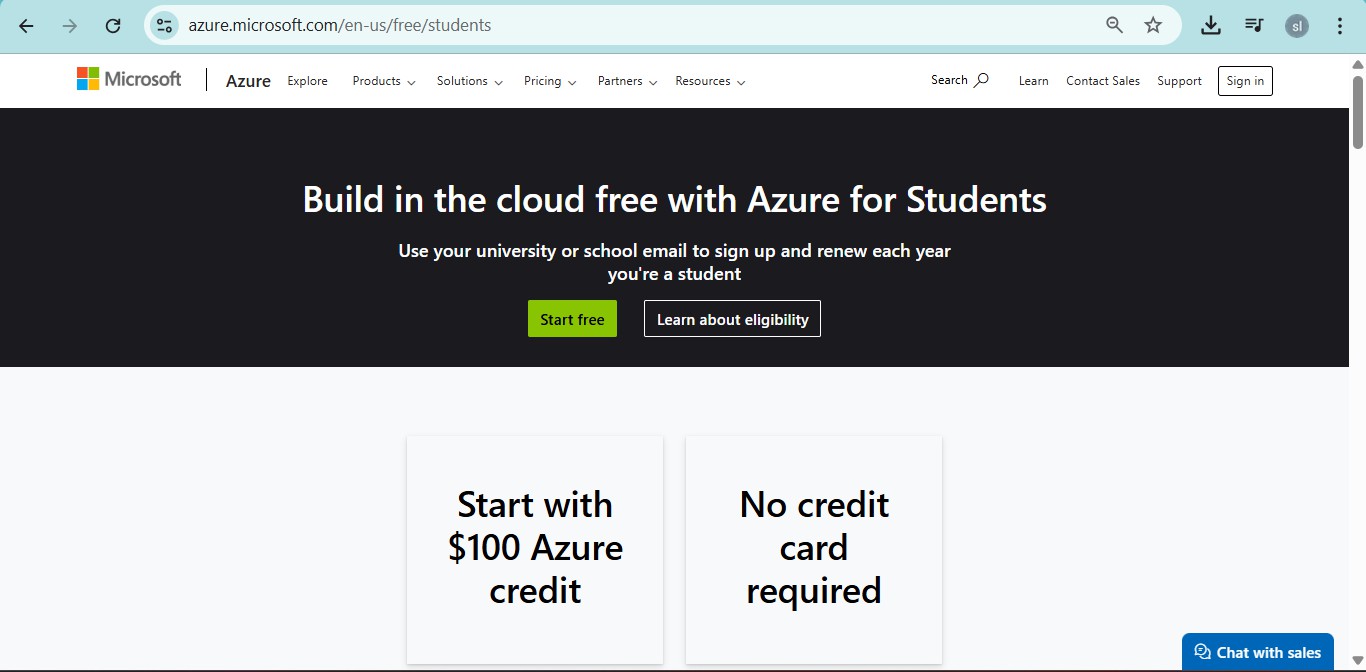
|  |  |
| --- | --- |
| **EXPNO:1** | **AZUREDEVOPSENVIRONMENTSETUP** |

## AIM

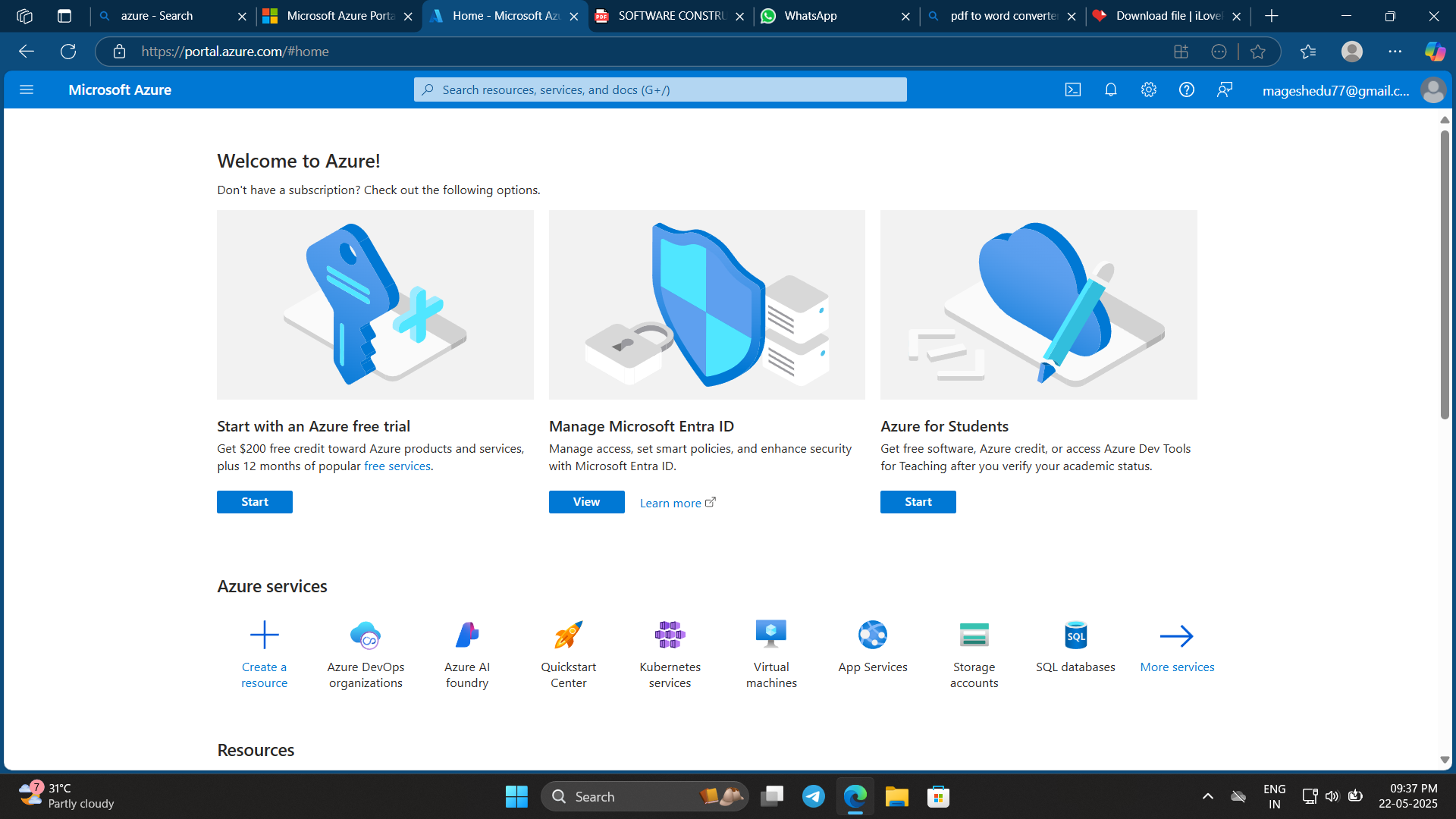
TosetupandaccesstheAzureDevOpsenvironmentbycreatinganorganizationthroughtheAzure portal.

## INSTALLATION

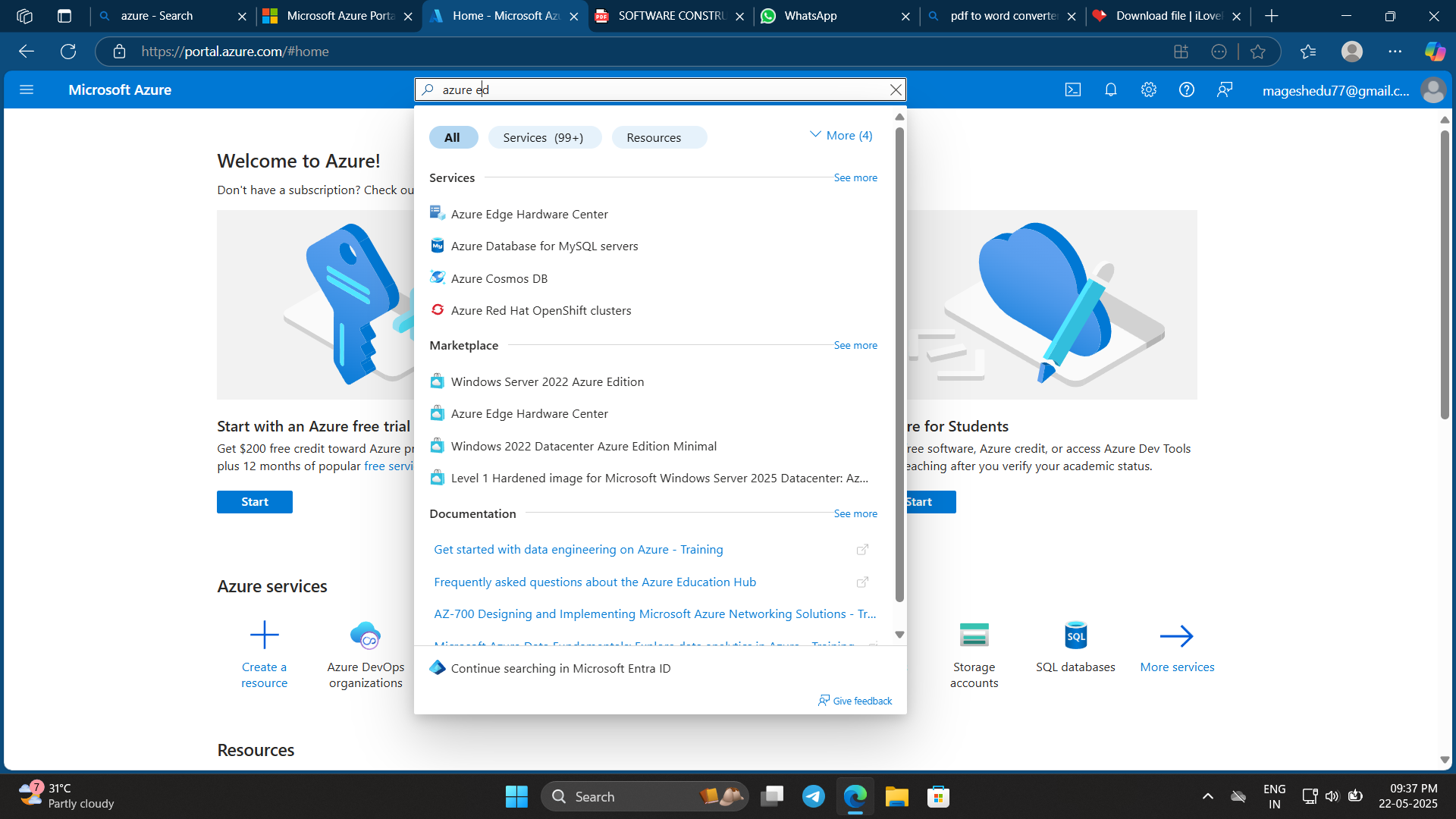
1. OpenyourwebbrowserandgototheAzurewebsite:[https://azure.microsoft.com/en-](https://azure.microsoft.com/en-us/get-started/azure-portal) [us/getstarted/azureportal.](https://azure.microsoft.com/en-us/get-started/azure-portal)

SigninusingyourMicrosoftaccount credentials. Ifyoudon'thaveaMicrosoftaccount,youcancreateonehere: <https://signup.live.com/?lic=1>

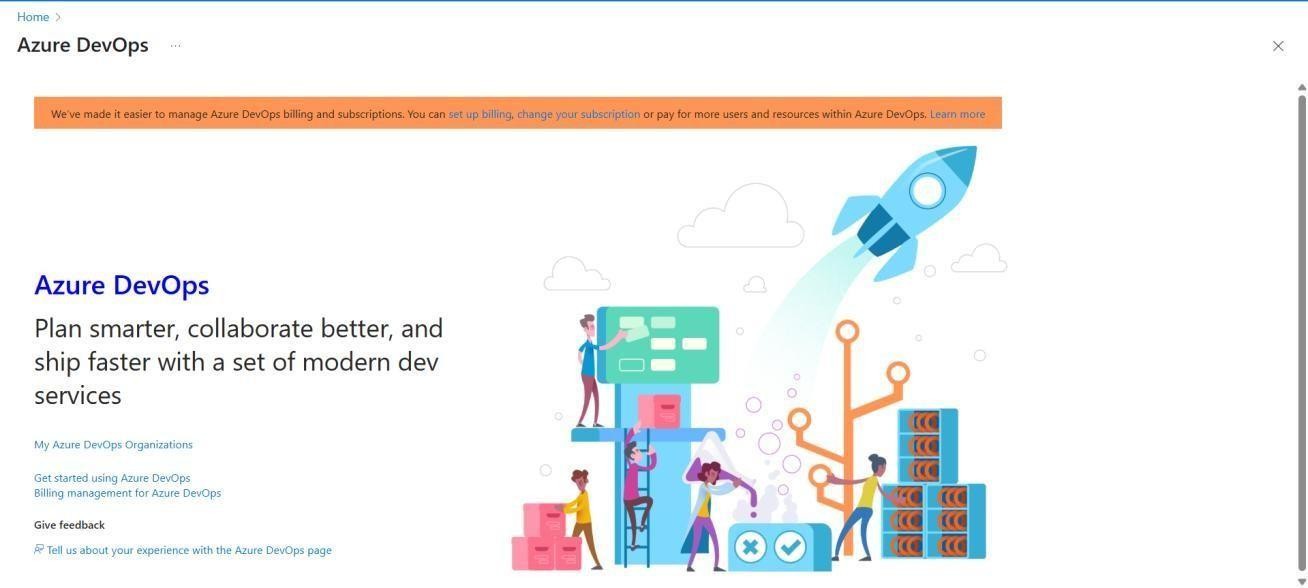
1. Azurehomepage



1. OpenDevOpsenvironmentintheAzureplatformbytyping ***AzureDevOpsOrganizations*** inthesearch bar.



1. Clickonthe***MyAzureDevOpsOrganization***linkandcreateanorganizationandyoushouldbetakento the Azure DevOps Organization Home page.



## RESULT

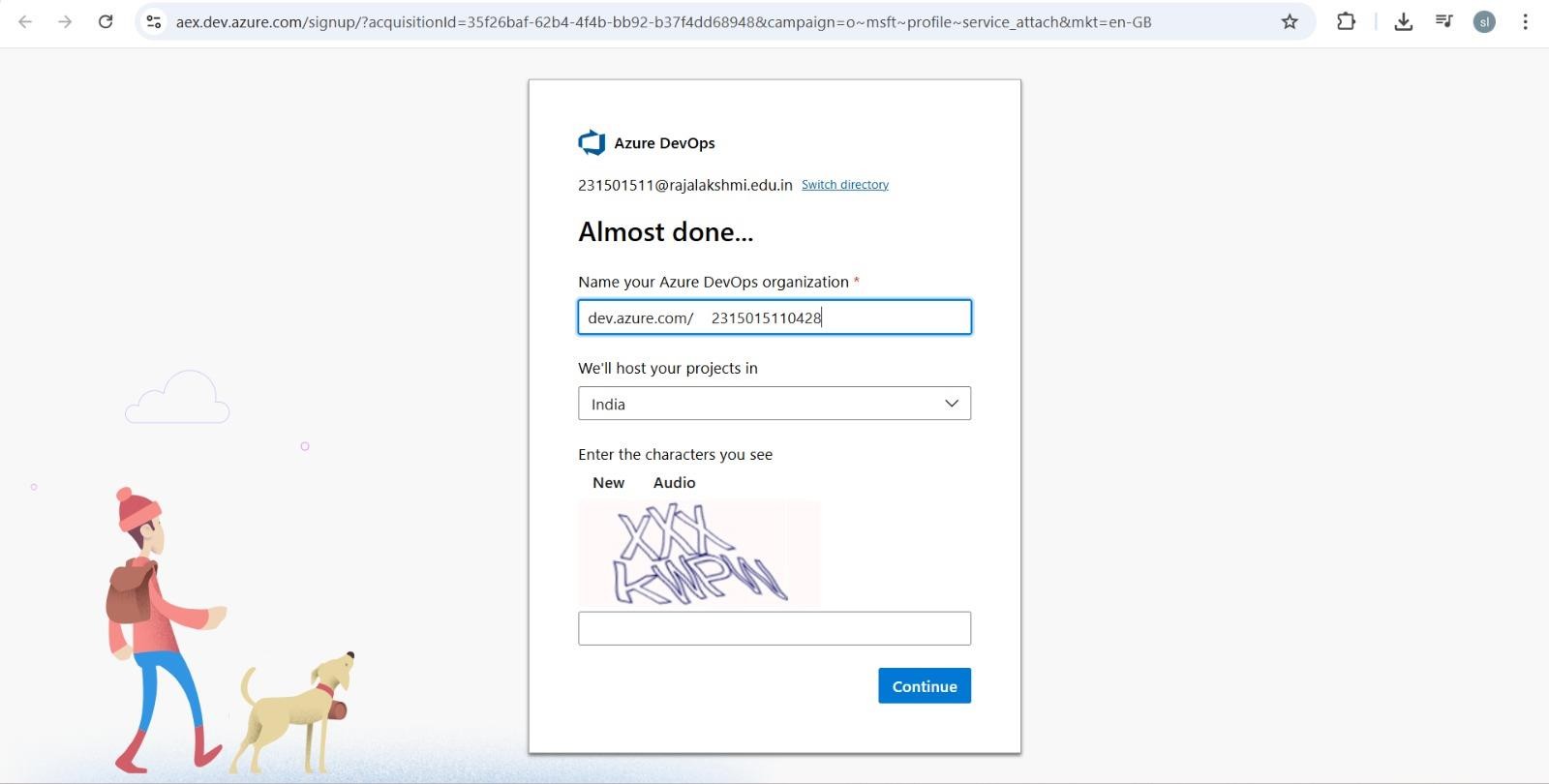
Successfully accessed the Azure DevOps environment and createdaneworganizationthroughthe Azure portal.

|  |  |
| --- | --- |
| **EXPNO:2** | **AZUREDEVOPSPROJECTSETUPANDUSERSTORY**  **MANAGEMENT** |

## AIM

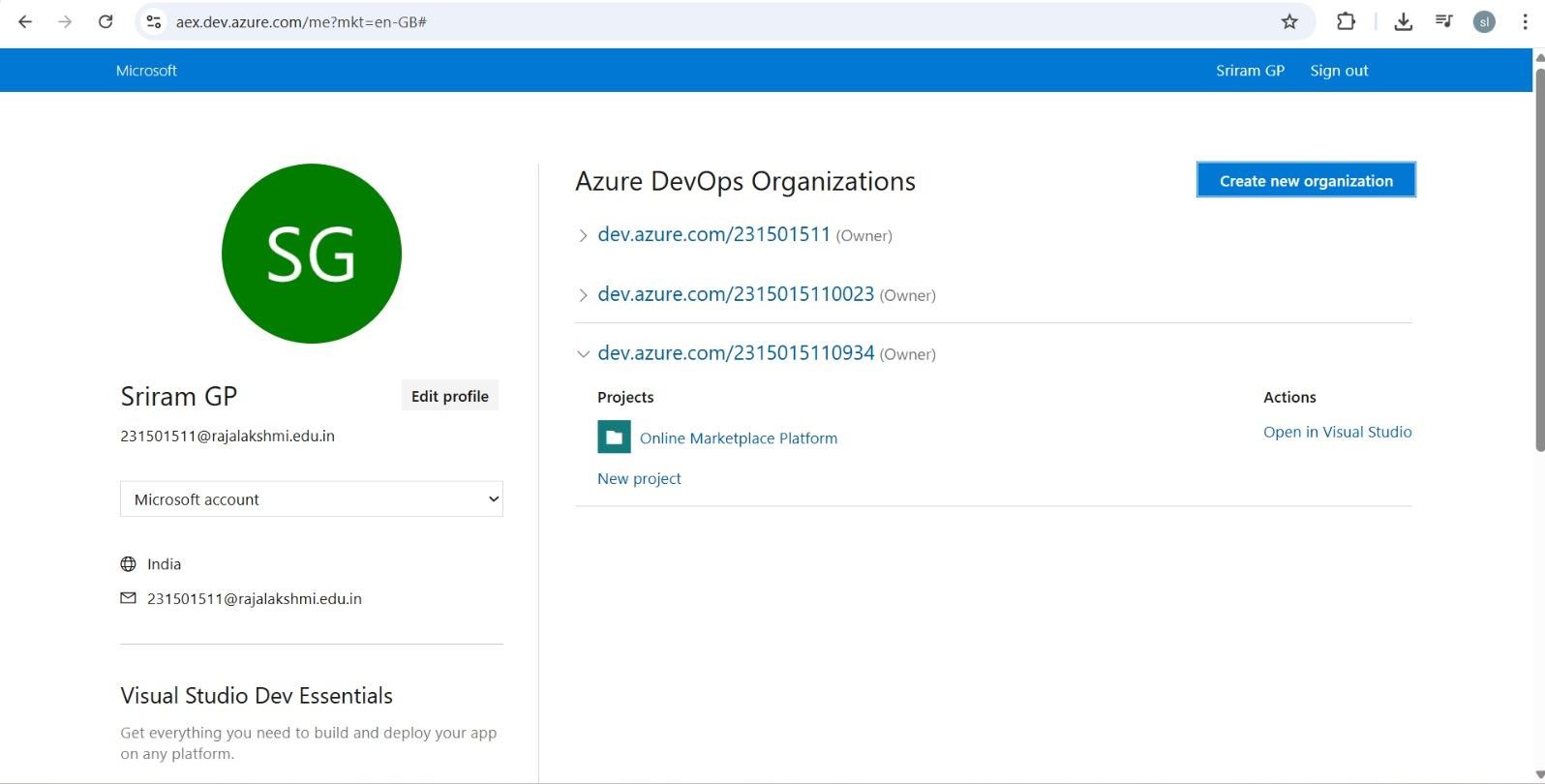
TosetupanAzureDevOpsprojectforefficientcollaborationandagileworkmanagement.

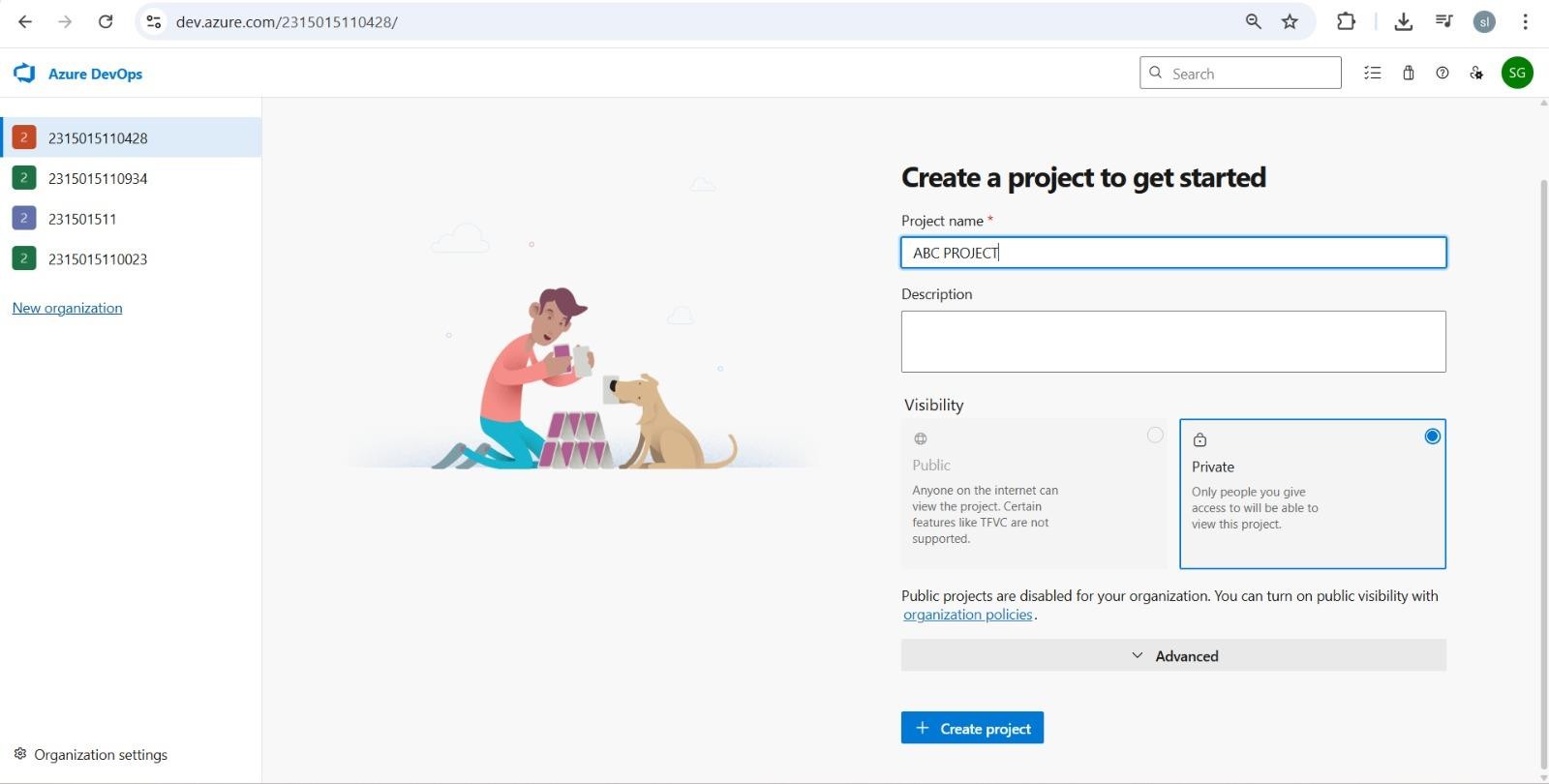
1. CreateAnAzureAccount



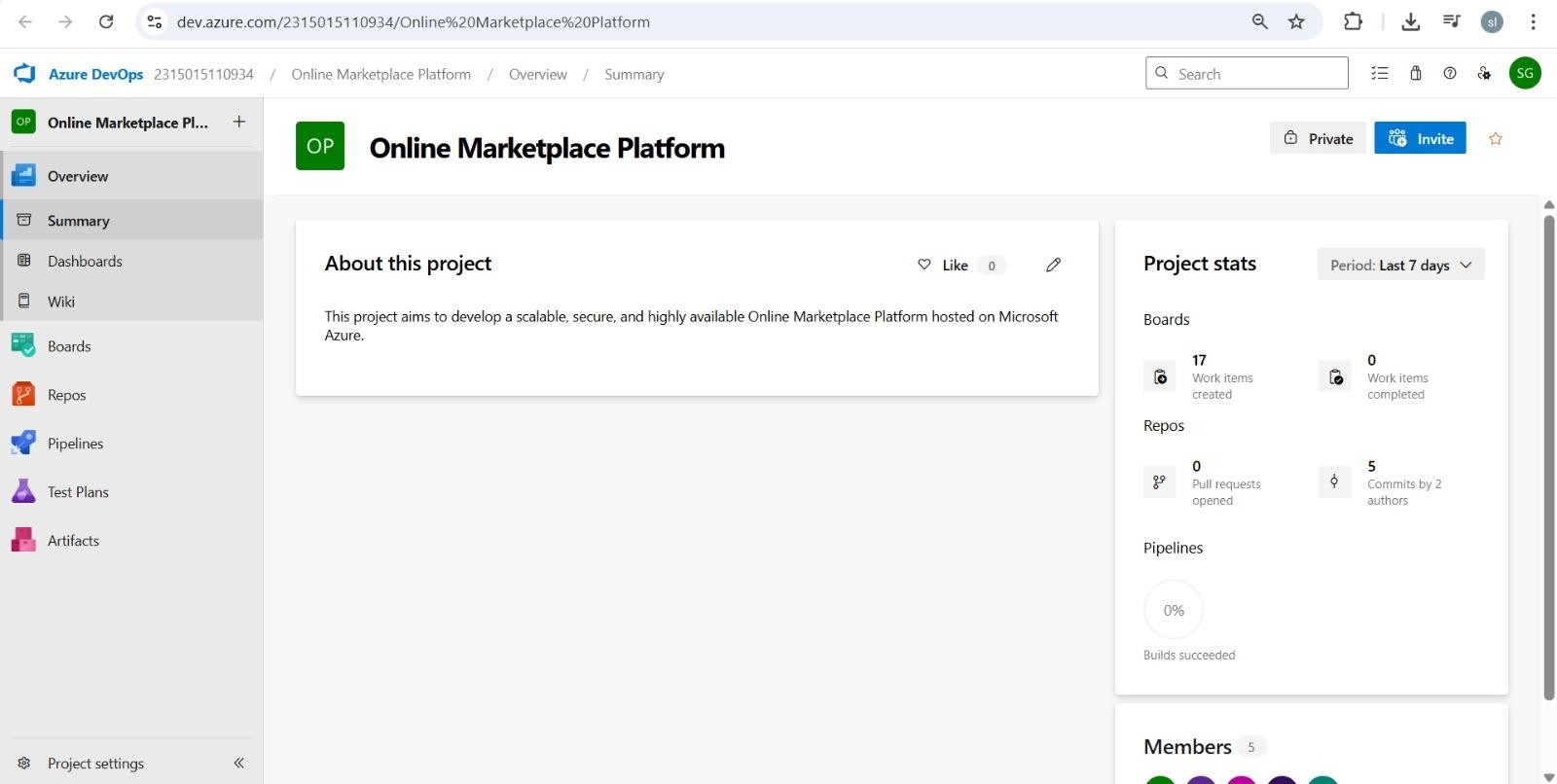
1. CreatetheFirstProjectinYourOrganization
   1. Aftertheorganizationissetup,you’llneedtocreateyourfirst **project**.Thisiswhereyou'll begin to manage code, pipelines, work items, and more.
   2. Ontheorganization’s**Homepage**,clickonthe**NewProject**button.
   3. Entertheprojectname,description,andvisibilityoptions:

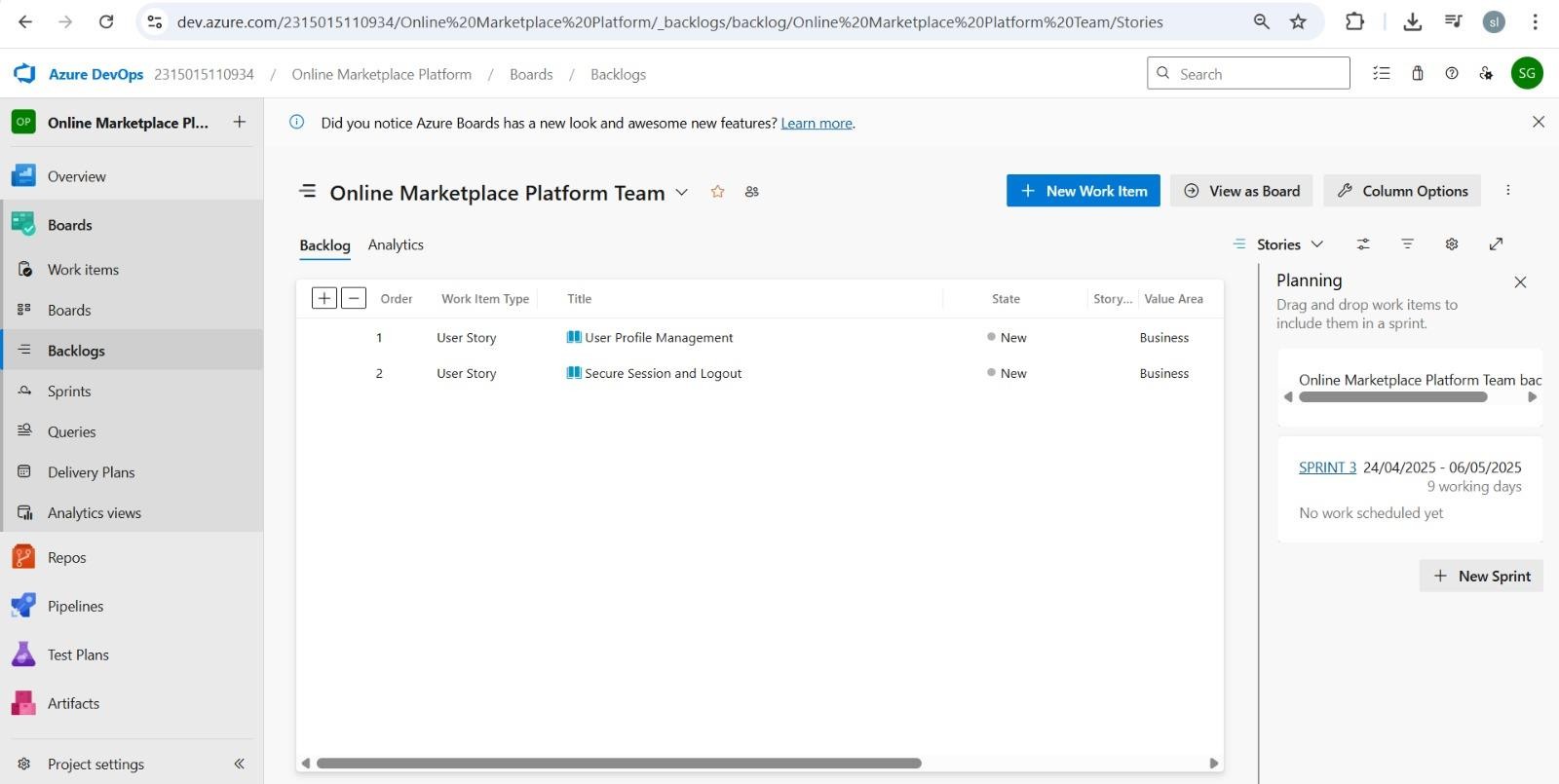
***Name****:*Chooseanamefortheproject(e.g.,LMS).

***Description****:* Optionally, add a description to provide more context about the project. ***Visibility****:*Choosewhetheryouwanttheprojecttobe**Private**(accessibleonlytothose invited) or **Public** (accessible to anyone).

* 1. Onceyou’vefilledoutthedetails,click**Create**tosetupyourfirst project.

1. Onceloggedin,ensureyouareinthecorrectorganization.Ifyou'repartofmultipleorganizations,you can switch between them from the top left corner (next to your user profile). Click on the Organization name, and you should be taken to theAzure DevOps Organization Home page.
2. Project dashboard



1. Tomanageuserstories:
   1. Fromthe**left-handnavigationmenu**,clickon**Boards**.Thiswilltakeyoutothemain**Boards** page,whereyoucanmanageworkitems,backlogs,andsprints.
   2. Onthe**workitems**page,you'llseetheoptionto**Addaworkitem**atthetop.Alternatively, youcanfinda**+**buttonor**AddNewWorkItem** dependingontheviewyou'rein.Fromthe **Addawork item** dropdown, select **User Story**. This will open a form to enter details for the new User Story.

.

## RESULT

setup.

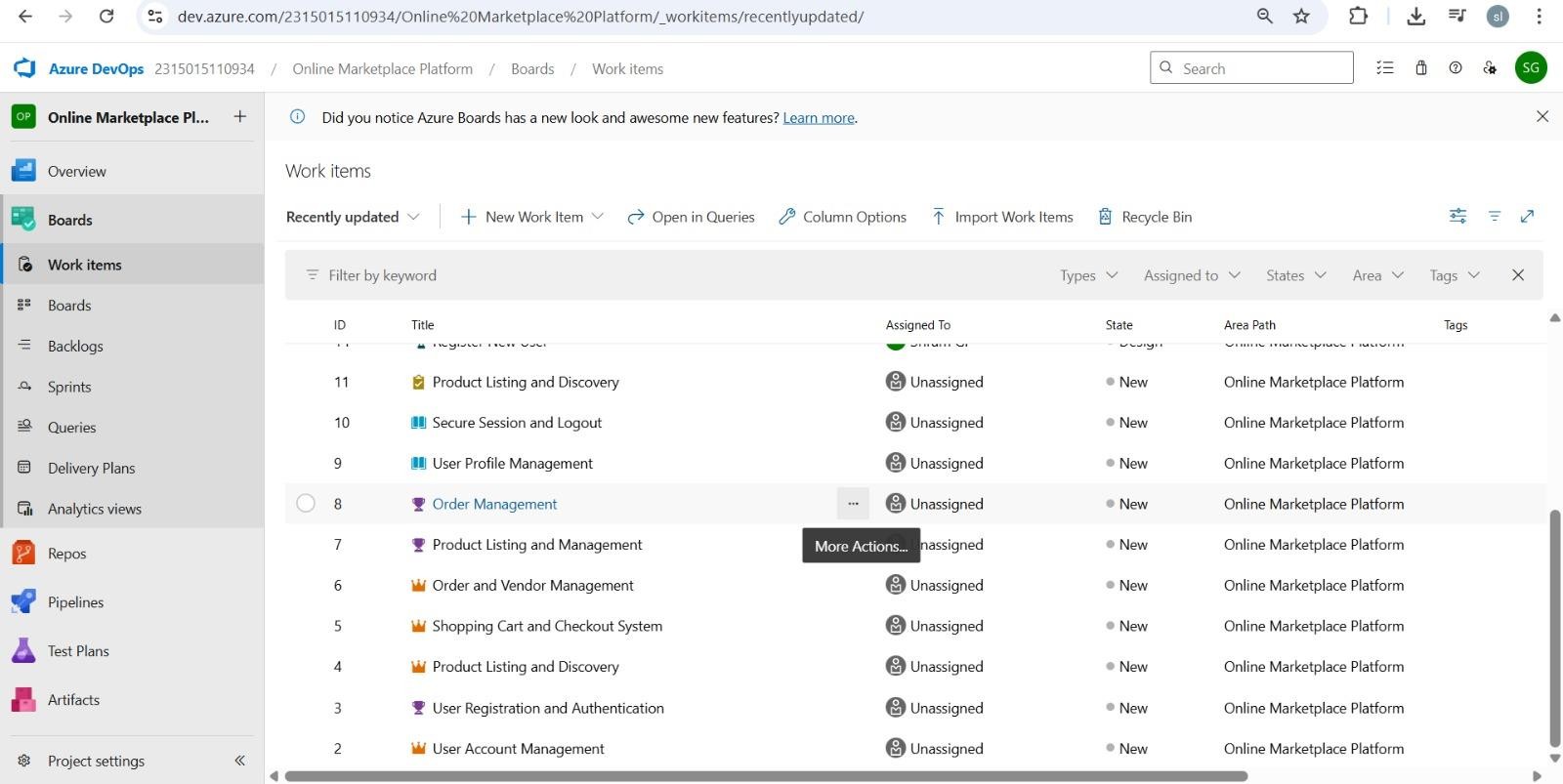
SuccessfullycreatedanAzureDevOpsprojectwithuserstorymanagementandagileworkflow

|  |  |
| --- | --- |
| **EXPNO:3** | **SETTINGUPEPICS,FEATURES,ANDUSERSTORIES**  **FORPROJECTPLANNING** |

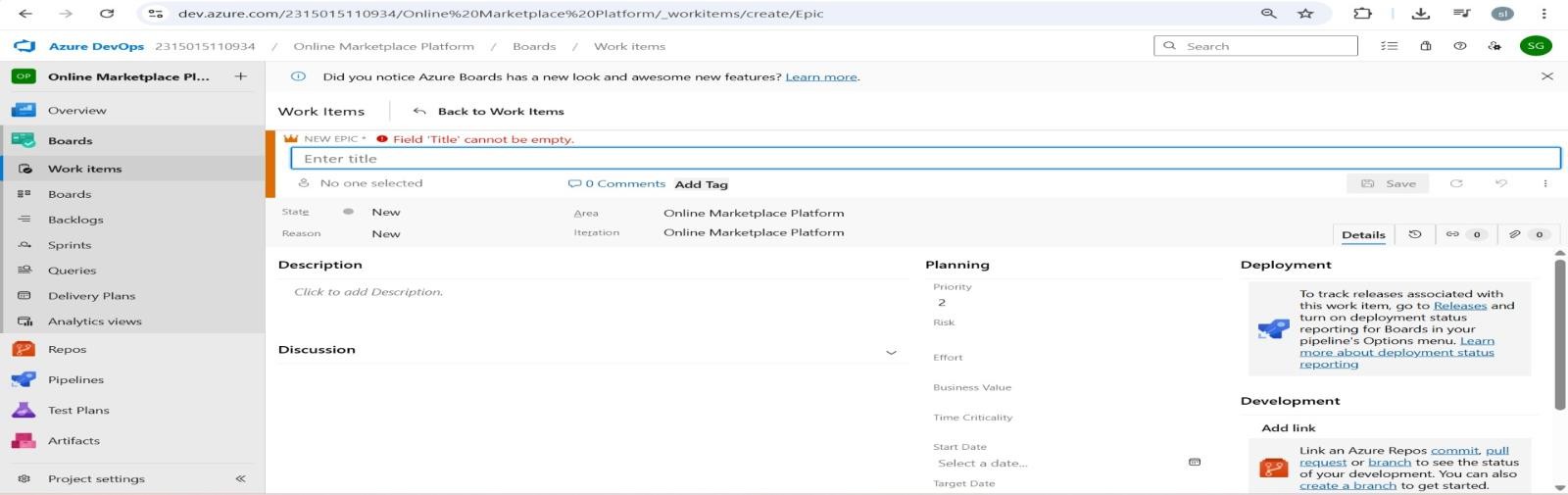
## AIM

To learn about how to create epics ,userstory ,features, backlogs for your assigned project.

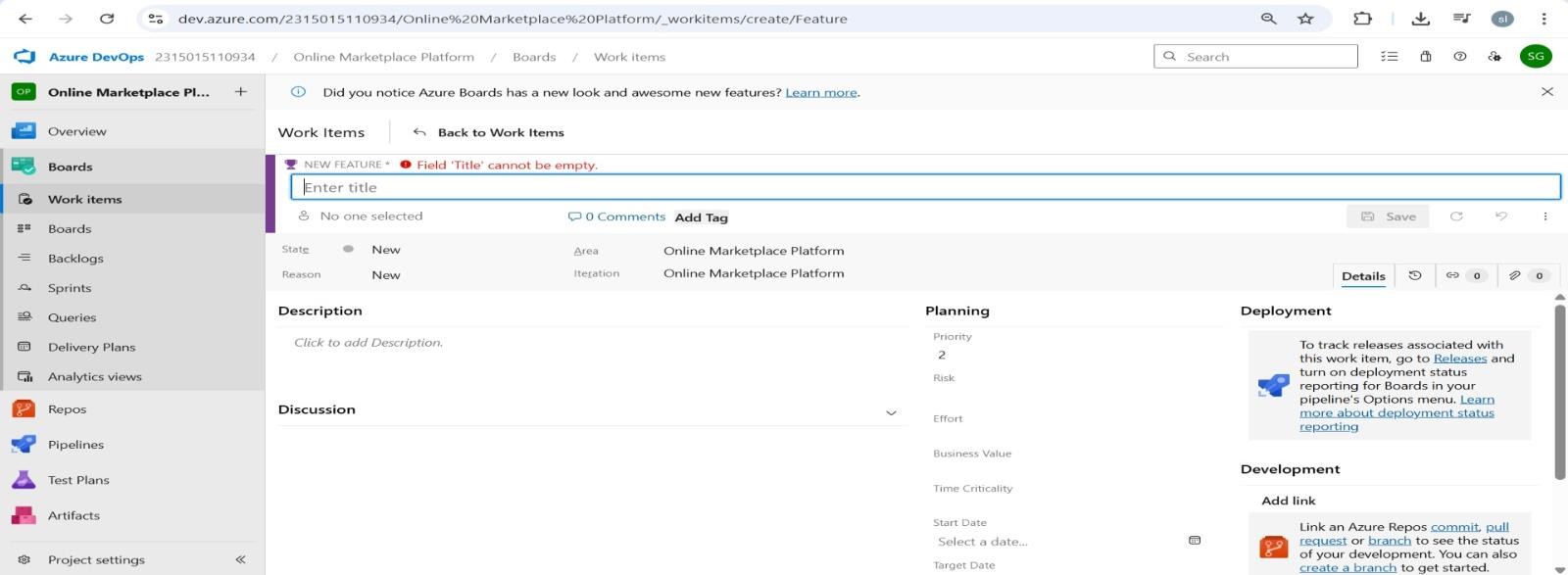
**CreateEpic,Features,UserStories,Task**

****

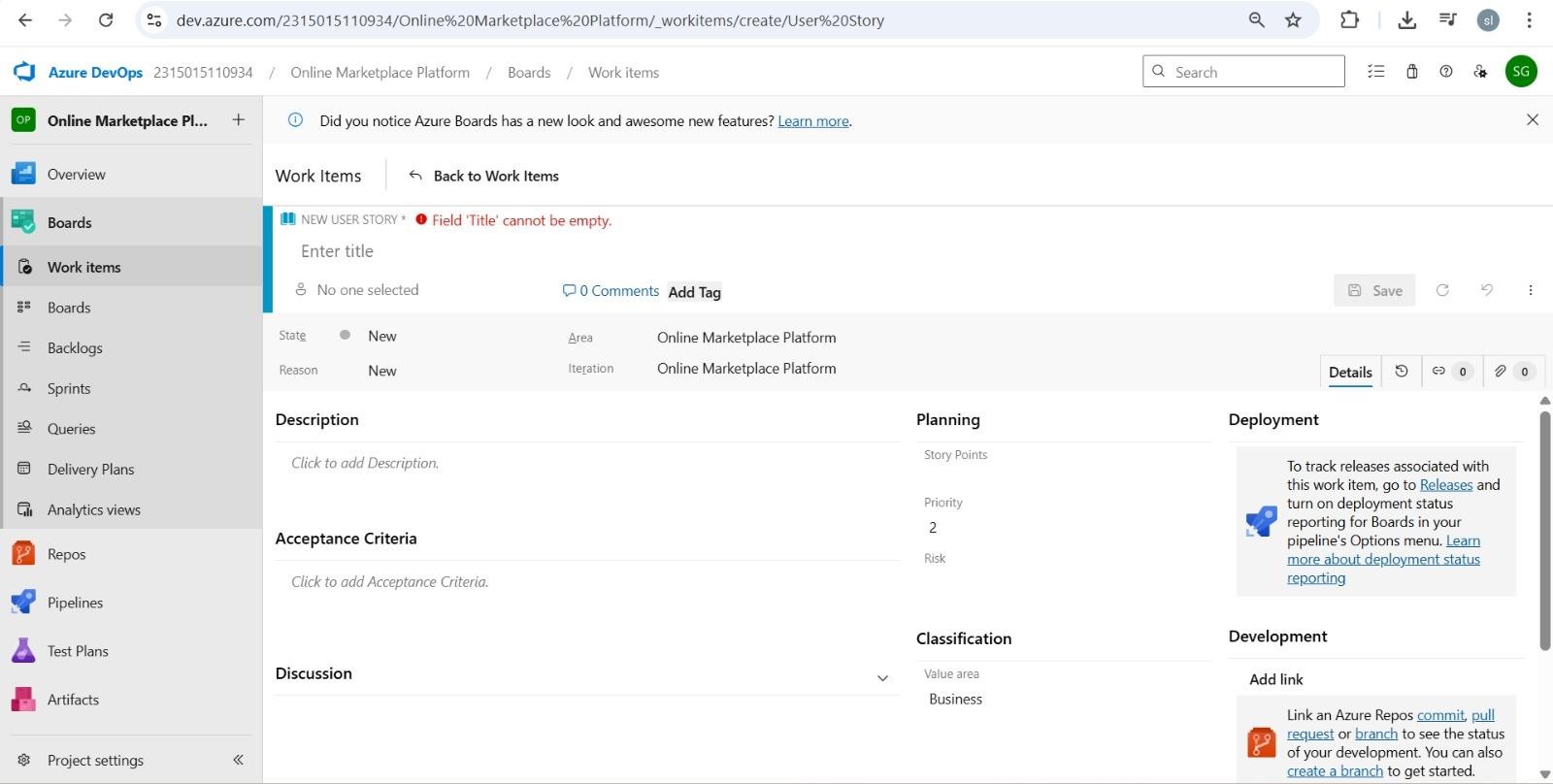
# Fill inEpics

****

1. **Fill inFeatures**

****

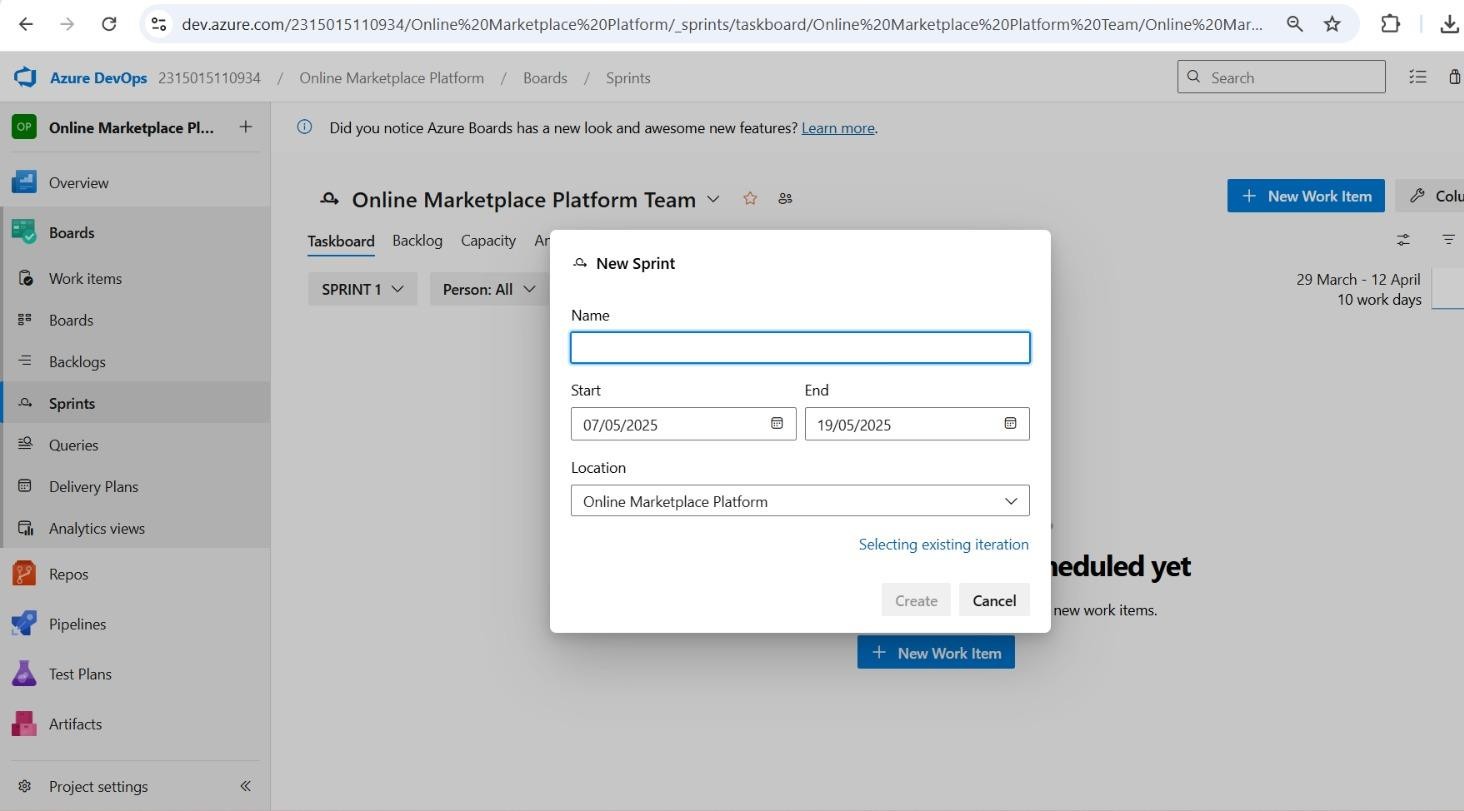
# Fillin UserStoryDetails

****

**RESULT**

Thus,thecreationofepics,features,userstoryand taskhasbeencreatedsuccessfully.

|  |
| --- |
| **SPRINTPLANNING** |
| **EXPNO:4** |

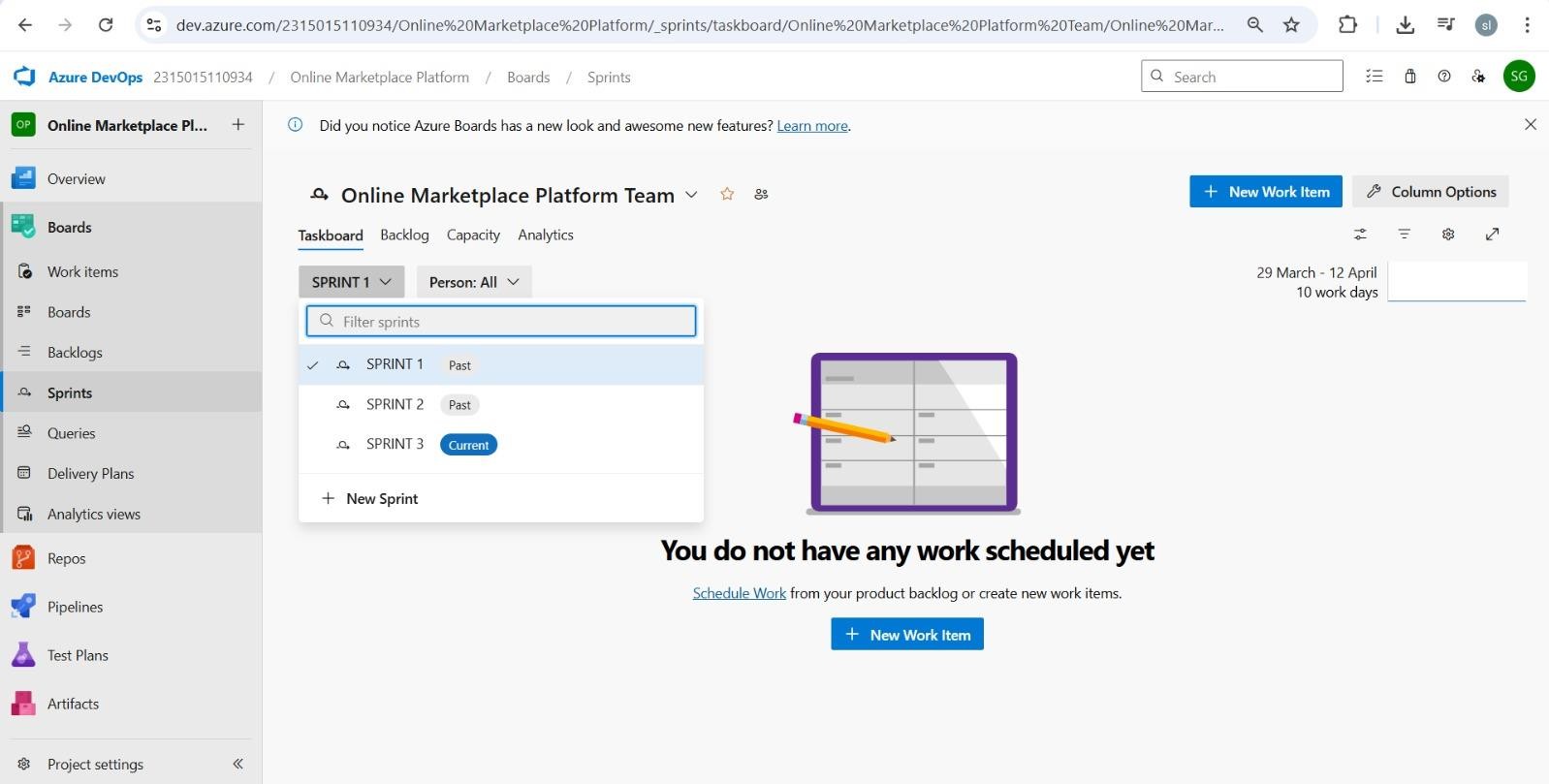


## AIM

To assign userstory to specific sprint for the Online market platform.

**Sprint CREATING**

**Sprint 1 2 3**

****

## RESULT

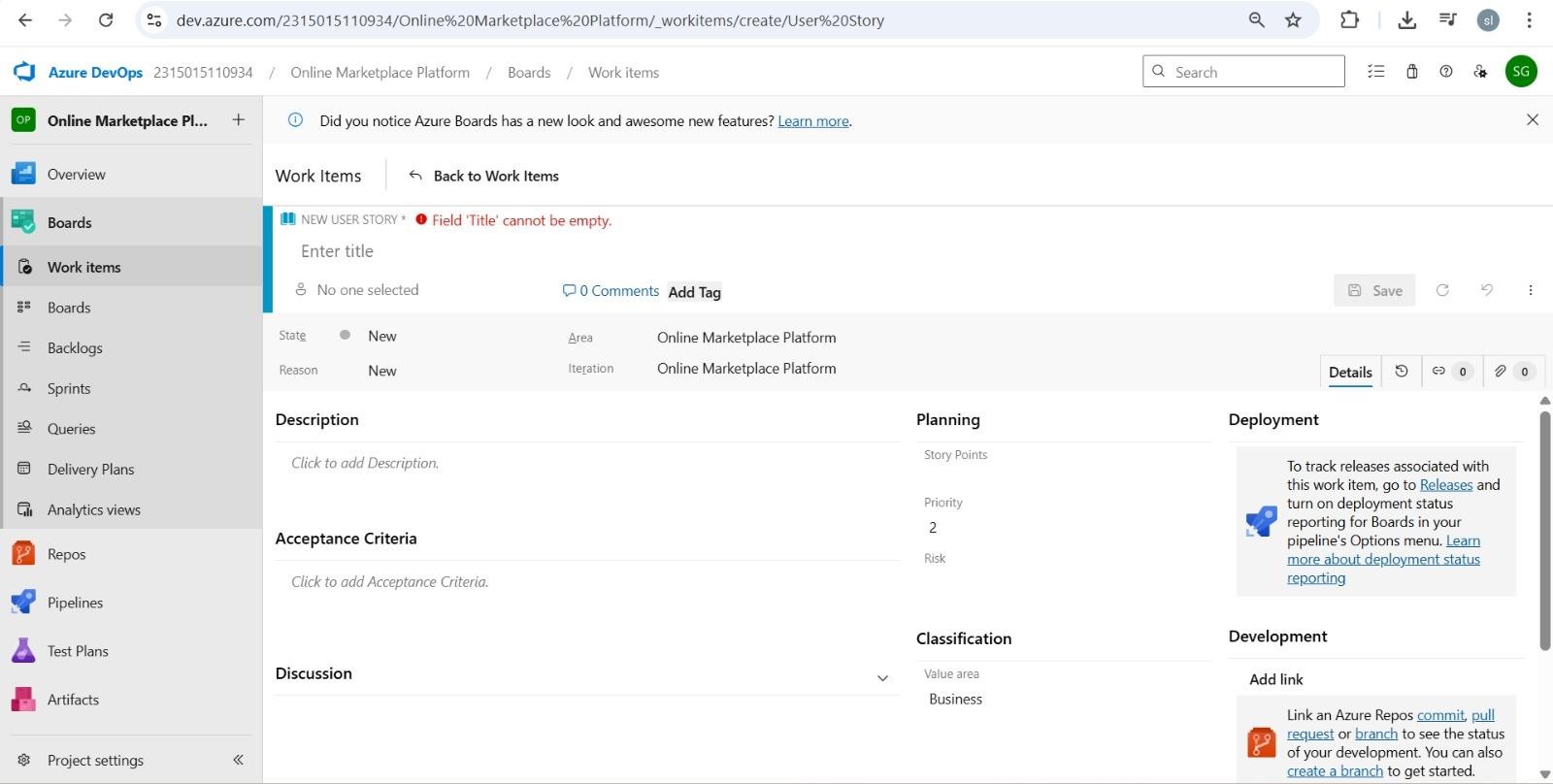
The Sprints have been created for Online Market Platform

|  |  |
| --- | --- |
| **EXPNO:5** | **POKERESTIMATION** |

## AIM

Create Poker Estimation for the user stories – Online Market Platform.

**PokerEstimation**

****

## RESULT

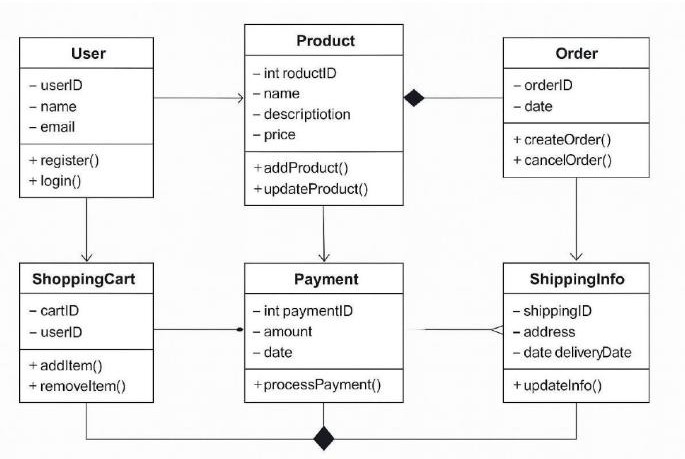
The Estimation/Story PointsiscreatedfortheprojectusingPoker Estimation.

|  |  |
| --- | --- |
| **EXPNO:6** | **DESIGNINGCLASSANDSEQUENCEDIAGRAMSFOR**  **PROJECTARCHITECTURE** |

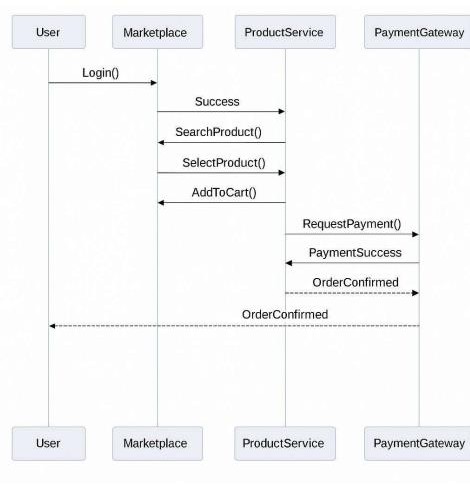
## AIM

To Designa Class Diagram and Sequence Diagram for the given Project.

**6A.Class Diagram**

****

**6B.SequenceDiagram**

****

## RESULT

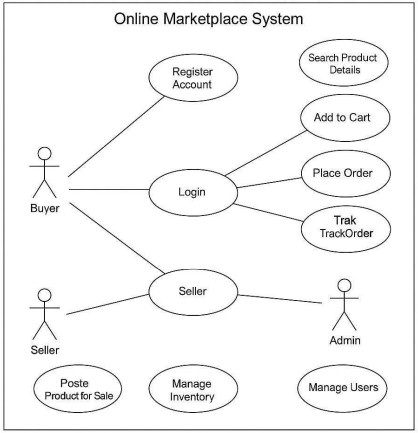
TheClassDiagramandSequenceDiagramisdesignedSuccessfullyfortheOnlineQuizSystem.

|  |  |
| --- | --- |
| **EXPNO:7** | **DESIGNINGUSE-CASEANDACTIVITYDIAGRAMS FOR PROJECT STRUCTURE** |

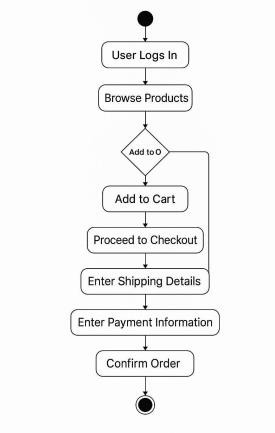
## AIM

ToDesignanUse-CaseDiagramandActivityDiagramforthegivenProject.**7A.**

**Use-CaseDiagram**

****

**7B.ActivityDiagram**

****

## RESULT

TheUse-CaseDiagramandActivityDiagramisdesignedSuccessfullyfortheOnlineQuizSystem.

|  |  |
| --- | --- |
| **EXPNO:8** | **TESTINGTESTPLANSANDTESTCASES** |

## AIM

TestPlansandTestCaseandwritetwotestcasesforatleastfiveuserstoriesshowcasing the

happypathanderrorscenariosinazureDevOps platform.

# TestPlanningandTestCaseTest Case Design Procedure

1. **UnderstandCoreFeaturesoftheApplication**
   * User Login
   * ​
2. **DefineUserInteractions**
   * Eachtestcasesimulatesarealuserbehaviour(e.g.,loggingin,submittingquizzes, viewing results)
3. **DesignHappyPathTestCases**
   * Focused on validating that all core functionalities work correctly under normalconditions
   * Example:Playerregistersandlogsin,submitsquizzesandviews results
4. **DesignErrorPathTestCases**
   * Simulateinvalidinputs,systemissuesorfailedactionstoensurepropererrorhandling.
   * Example: Login with invalid credentials, submission without attachments, unauthorized access to admin panel.
5. **BreakDownStepsandExpectedResults**
   * Eachtestcaseincludesaclearsequenceofactionsandexpectedresults.• Ensuresbothmanualtestersandautomationtoolscanfollowtheprocess

easily.

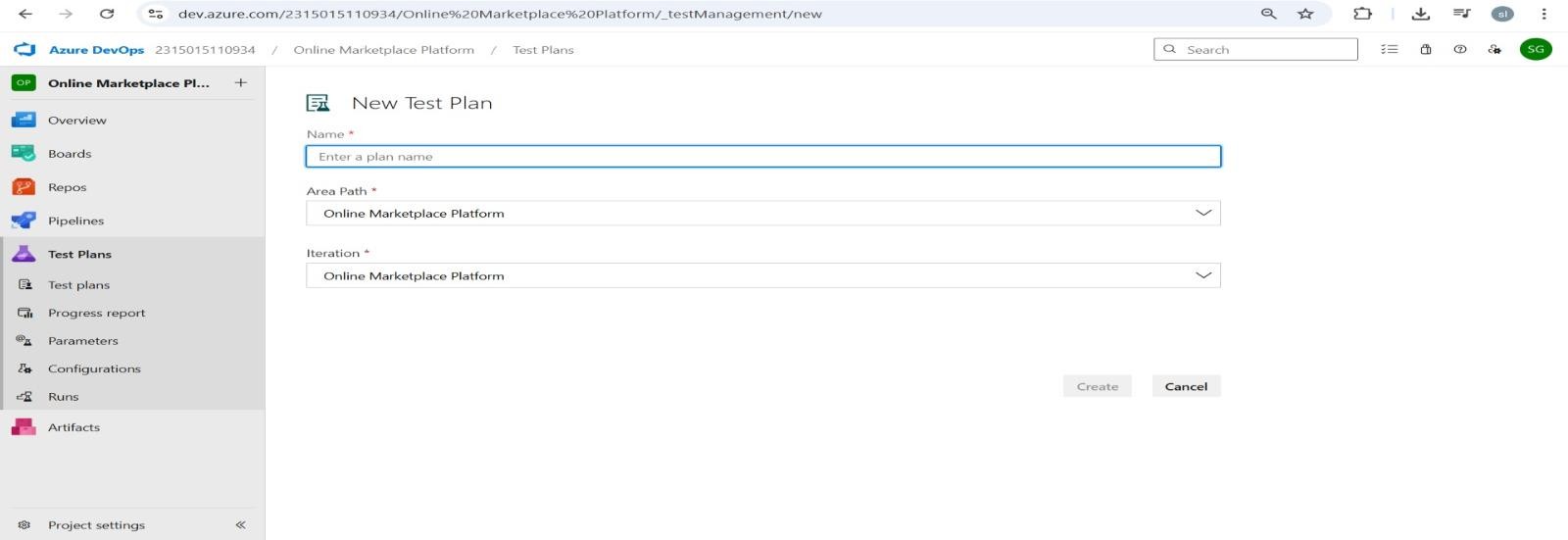
1. **UseClearNamingandIDs**
   * Testcasesareuniquelyidentifies(e.g.,TC01–ValidLogin,TC03– Invalid Password). • Facilities easy mapping to features and tracking in Azure DevOps.
2. **SeparateTest**
   * Groupedbyfunctionalitysuchas:

oLogin and Registration oQuizzes Submission oViewing Results oAdmin Functions • Improvesorganizationandenablesfocused execution in Azure DevOps.

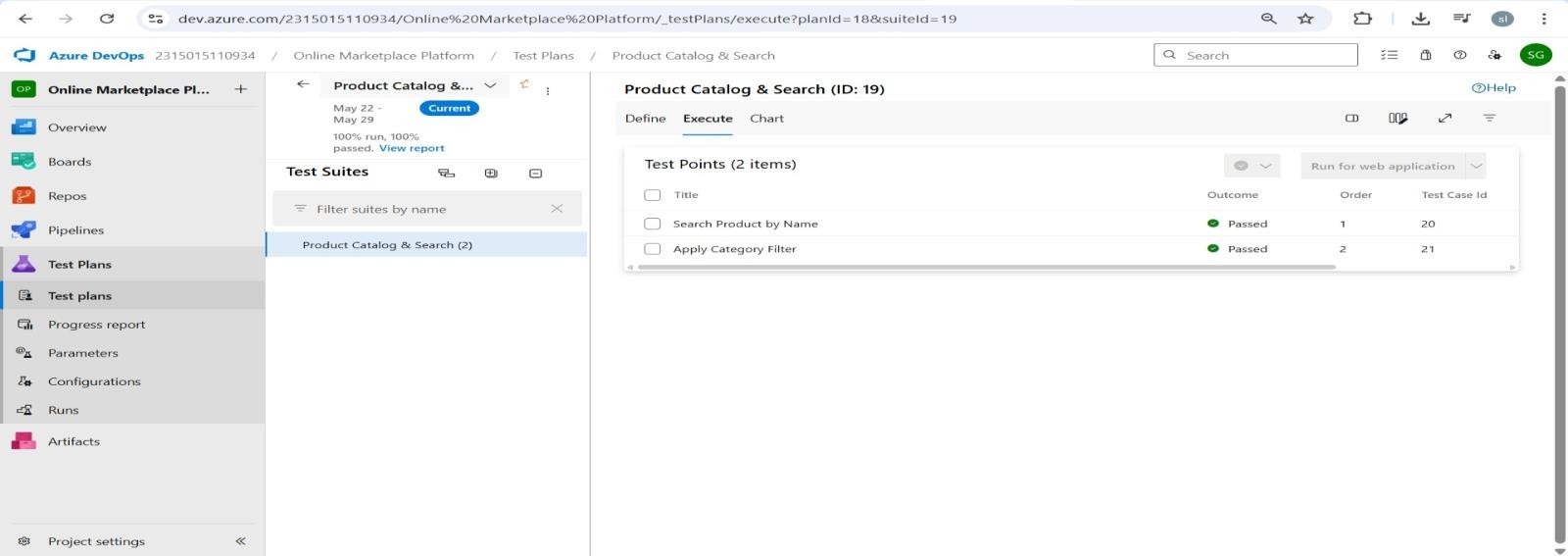
1. **PrioritizeandReview**
   * High-priorityassignedtocriticalworkflowslikelogin,quizzesandresults.
   * Reviewedforcompleteness,accuracyandalignmentwithuserstoriesandfeatures definition.

.

1. **New test plan**

****

1. **Testsuite**

****

1. **Testcase**

Givetwotestcasesforatleastfiveuserstoriesshowcasingthehappypathanderrorscenariosinazure DevOps platform.

Online market platform –Test Plans

## USERSTORIES

* + Asanadmin,Iwanttologintomanagequizzesandusers(ID: 3).
  + Asaplayer,IwanttoregisterandloginsothatIcanaccessquizzessecurely(ID:4).
  + Asanadmin,Iwanttogiveuserstherightaccesssotheycanusethesystemproperly(ID:6).
  + Asaplayer,Iwanttoseeonlymyquizzesandprogresssothatit’seasytouse(ID:7).
  + Asanadmin,IwanttocreateandconfigurequizzeswithtimelimitssoIcancontrolquizflow (ID: 10).

## TESTSUITES

**TestSuite: TS01**

**-User Registration & Login\* TC01–Register New User**

* + - **Action:**

 Navigate to Sign-Up page

 Enter valid details (Name, Email, Password

 Click "Register"

* + - * **ExpectedResults:**

 User is registered and redirected to Dashboard

* + - * **Type**:HappyPath

1. **TC02–Login with Valid Credentials**
   * **Action:**

 Navigate to Login page

 Enter valid email and password

* + **ExpectedResults:**

 user is logged in and redirected to homepage

.

**TestSuite:TS02–**

**Product Catalog & Search\***

**TC03–Search Product by Name**

* **Action:**

 Enter product name in search bar

 Click search icon

* **ExpectedResults:**

 Relevant products are displayed

**TC04–Apply Category Filter**

* **Action:**

 Navigate to product listing

 Select a category filter<br>3. Observe results

* **ExpectedResults:**

 Only products from selected category are shown

.

**TestSuite:TS03–**

**Cart & Checkout\***

**TC05 Add Product to Cart**

* **Action:**

 Open product detail<br>2. Click "Add to Cart"

* **ExpectedResults:**

 Product is added to cart, cart count increases.

1. **TC06–Complete Checkout with Credit Card**
   * **Action:**

 Add product to cart

 Proceed to checkout

Enter valid credit card details

. Click Pay

o **ExpectedResults:**

 Order is placed, confirmation is shown

**TestSuite:TS04–**

**API Testing (Using Postman)\***

1. **TC07–GET /products**

* **Action:**

 Send GET request to /api/product

* **ExpectedResults:**

 200 OK with product JSON list |.

**TestSuite:TS05– Performance Testing**

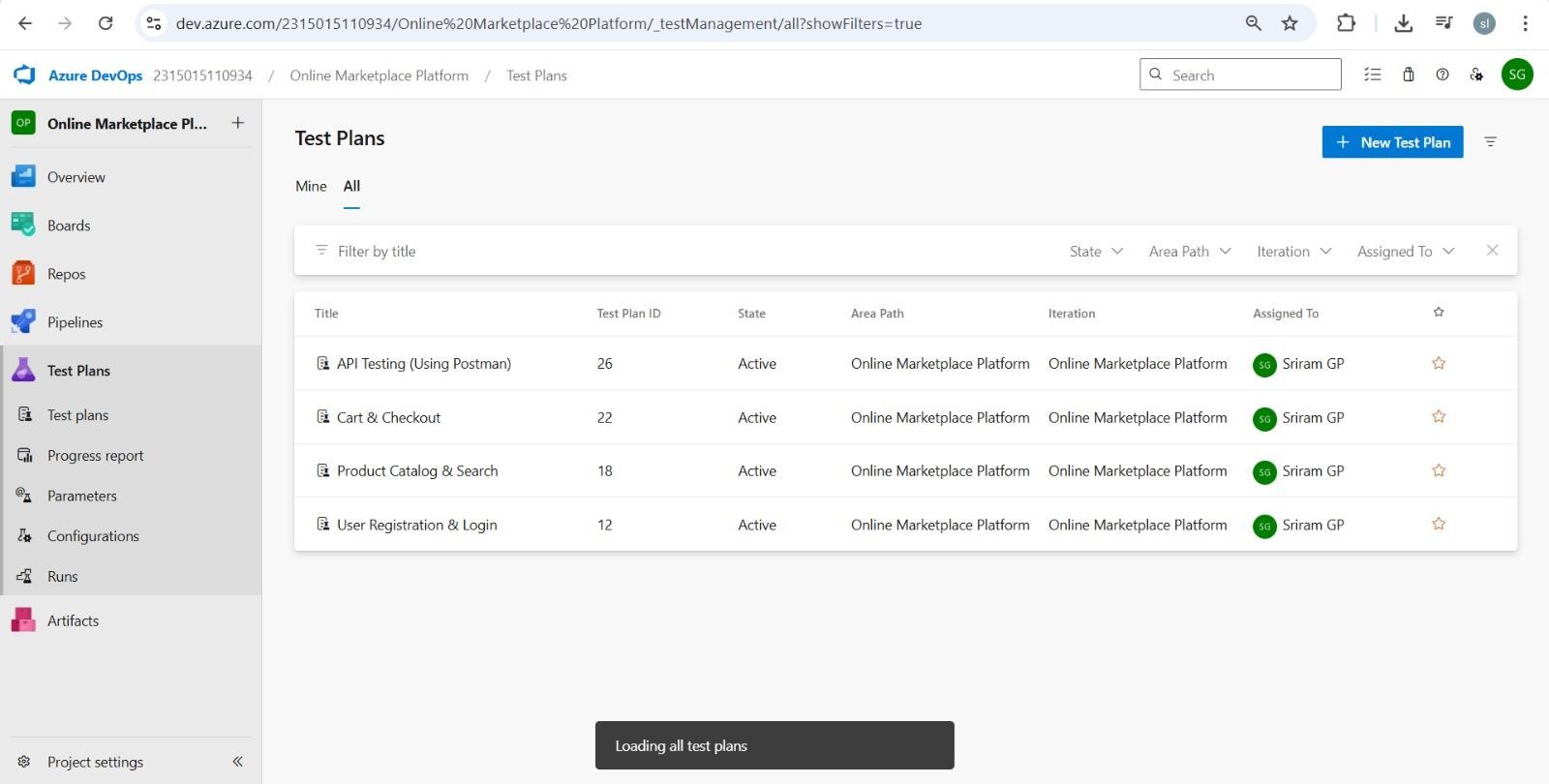
**1. TC08–Homepage Load Test o Action:**

 Simulate 1000 users accessing homepage concurrently

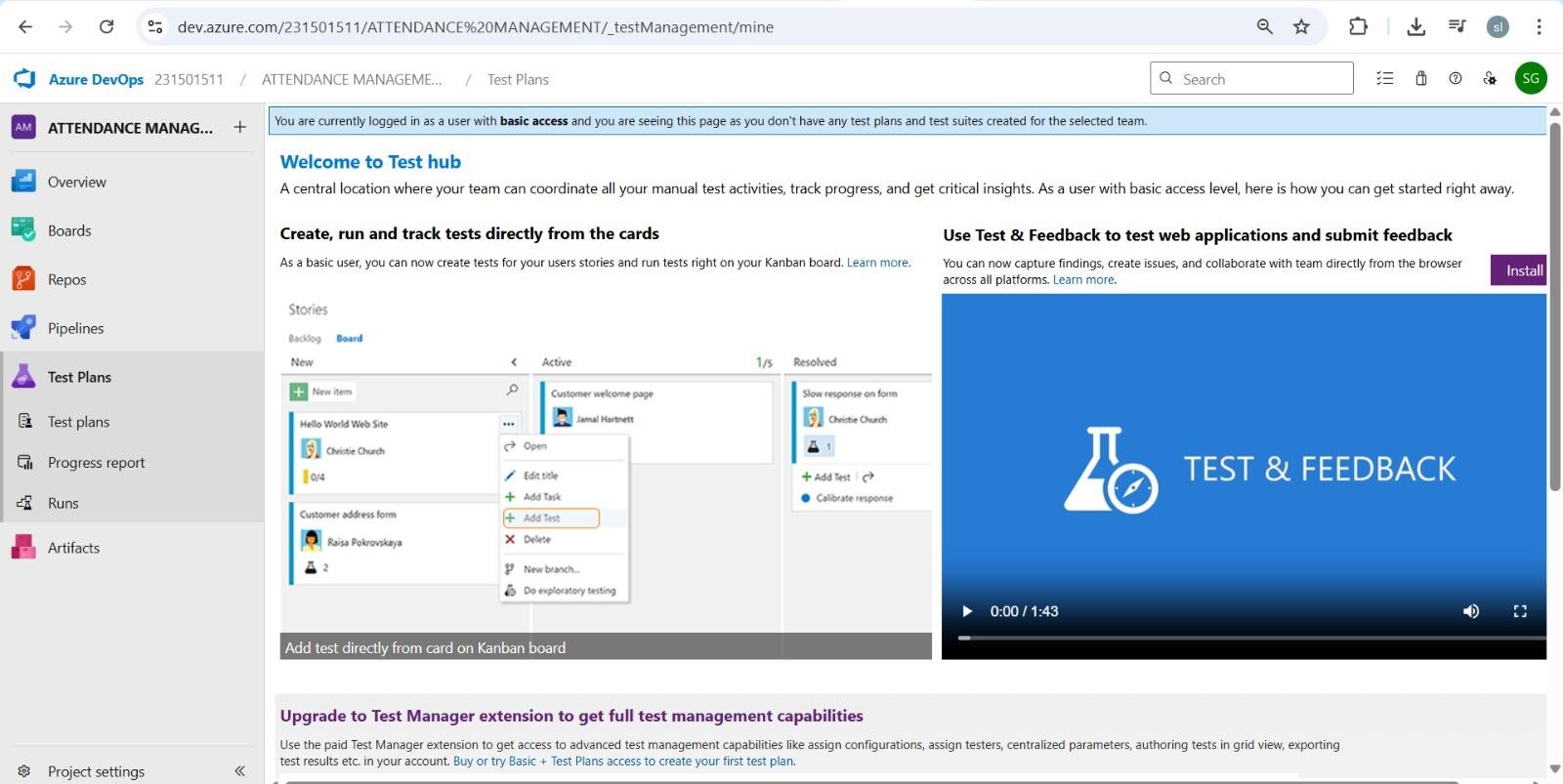
**o ExpectedResults:**

 Load time under 3 seconds, no crashes

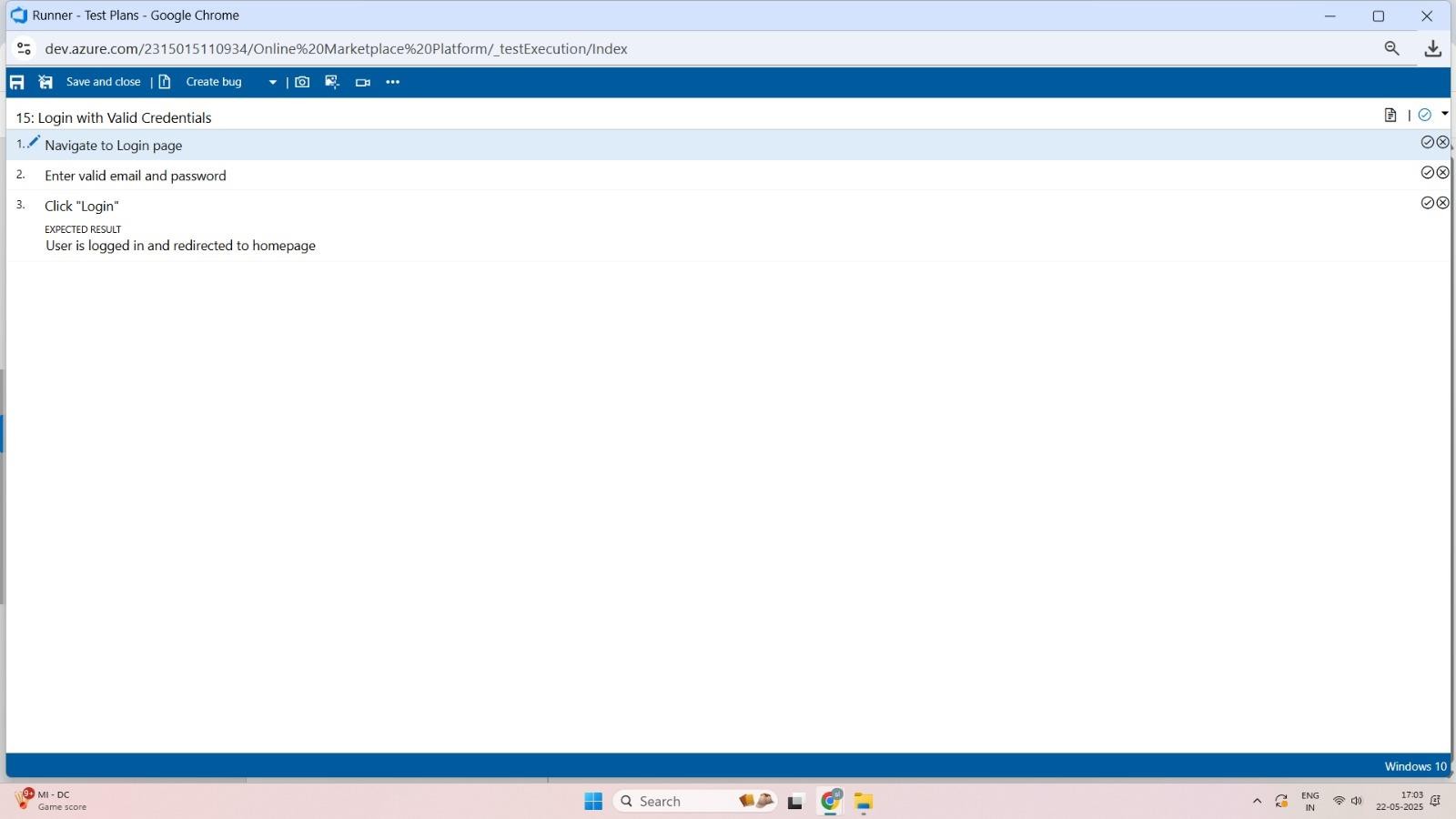
**TestCases**

****

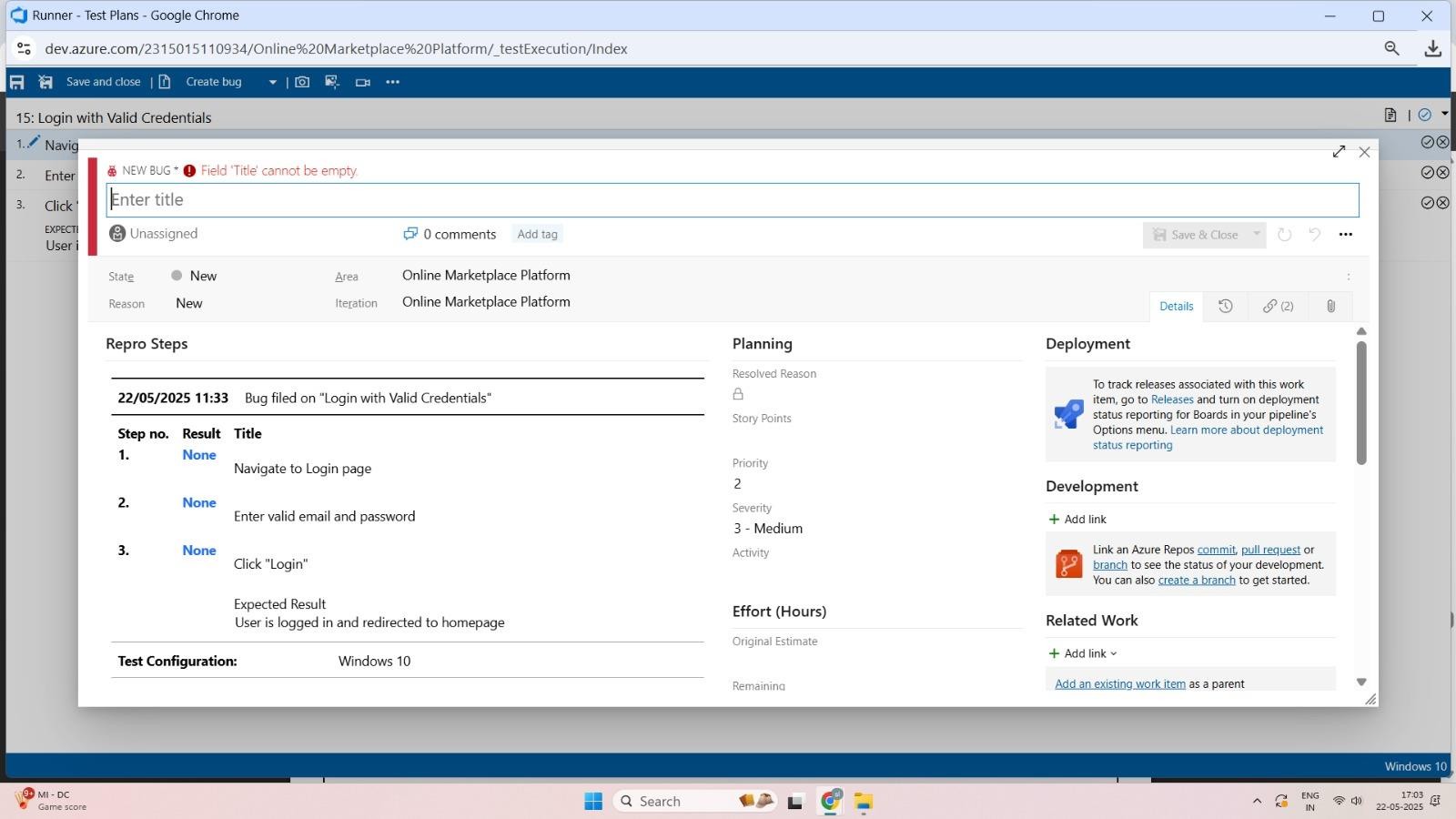
**Installationof test**

****

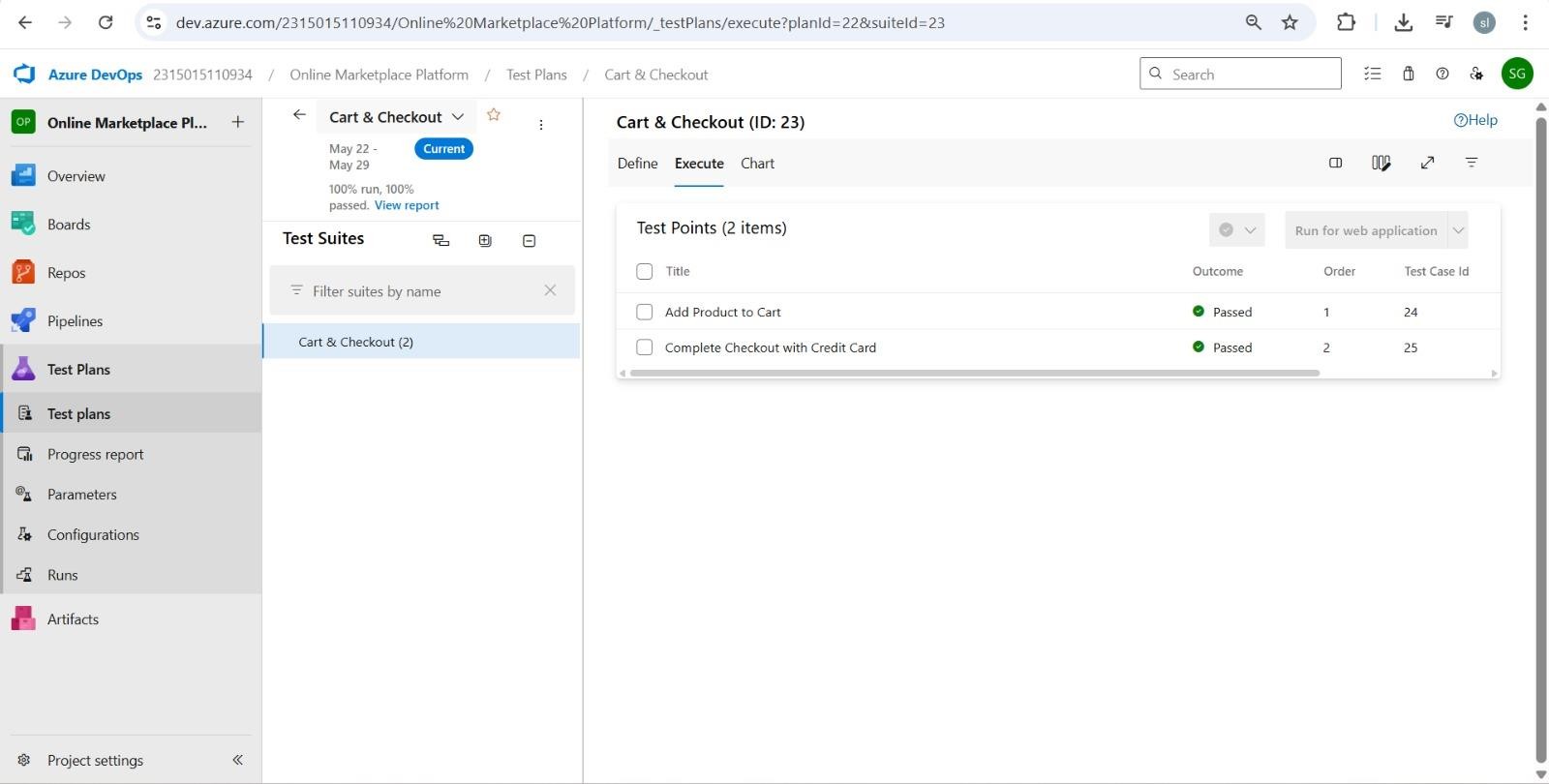
**Runningthetestcases**

****

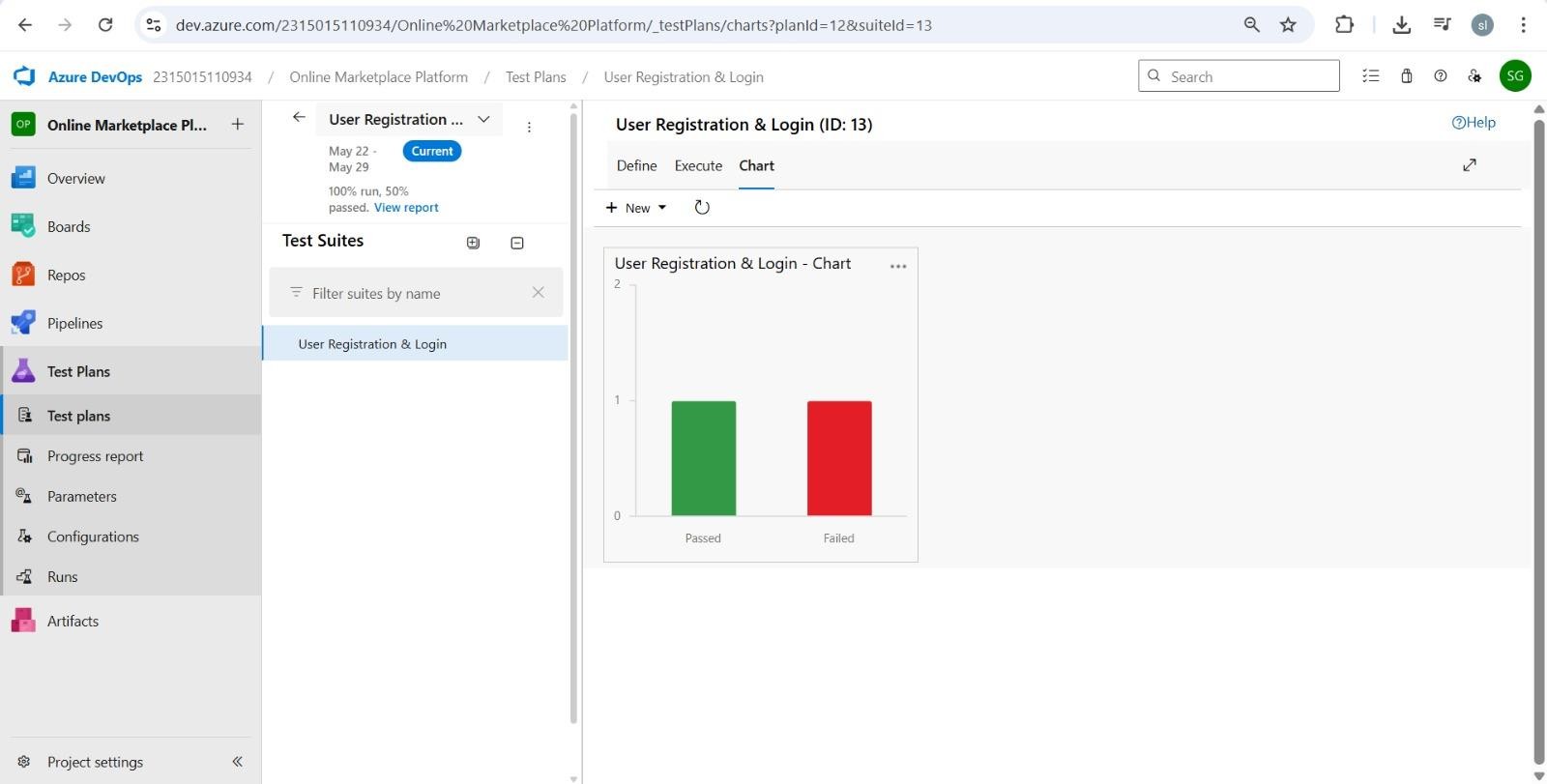
**Creating the bug**

****

**Testcaseresults**

****

**Progress report**

****

## RESULT

Path.

ThetestplansandtestcasesfortheuserstoriesiscreatedinAzureDevOpswithHappyPathand Error

|  |  |
| --- | --- |
| **EXPNO:9** | **CI/CDPIPELINESINAZURE** |

## AIM

To implement a Continuous Integration and Continuous Deployment (CI/CD) pipeline inAzure DevOpsforautomatingthebuild,testing,anddeploymentprocessoftheStudentManagementSystem, ensuring faster delivery and improved software quality**.**

## PROCEDURE

**StepstoCreateandimplementpipelinesinAzure:**

1. SignintoAzureDevOpsandNavigatetoYourProject Loginto[dev.azure.com,](https://dev.azure.com/)selectyourorganization,andopentheprojectwhereyour Student Management System code resides.
2. ConnectaCodeRepository(AzureReposorGitHub)

Ensure your application code is stored in a Git-based repository such asAzure Repos or GitHub. This will be the source for triggering builds and deployments in your pipeline.

1. CreateaNew Pipeline

GotothePipelinessectionontheleftpaneland click“CreatePipeline”.

Chooseyoursource(e.g., AzureReposGitorGitHub),andthenselecttherepository containing your project code.

1. Choosethe PipelineConfiguration

YoucanselecteithertheYAML-basedpipeline(recommendedforversioncontroland automation) or the Classic Editor for a GUI-based setup. If usingYAML,Azure DevOps will suggest a template or allow you to define your own.

1. DefineBuildStage(CI-ContinuousIntegration)fromYAMLfile.
2. Installdependencies(e.g.,npminstall,dotnet restore).
3. Buildtheapplication(dotnet build,npmrun build).
4. Rununittests(dotnettest,npmtest).
5. Publishbuildartifactstobeusedinthereleasestage.
6. Saveand Runthe Pipelineforthe FirstTime SavetheYAMLorbuilddefinitionandclick“Run”.

Azurewillfetch thelatest codeandexecutethedefinedbuild andtest stages.

1. ConfigureContinuousDeployment(CD)

Navigate to the Releases tab under Pipelines and click “New Release Pipeline”. Add an Artifact (from the build stage) and create a new Stage (e.g., Development, Production).

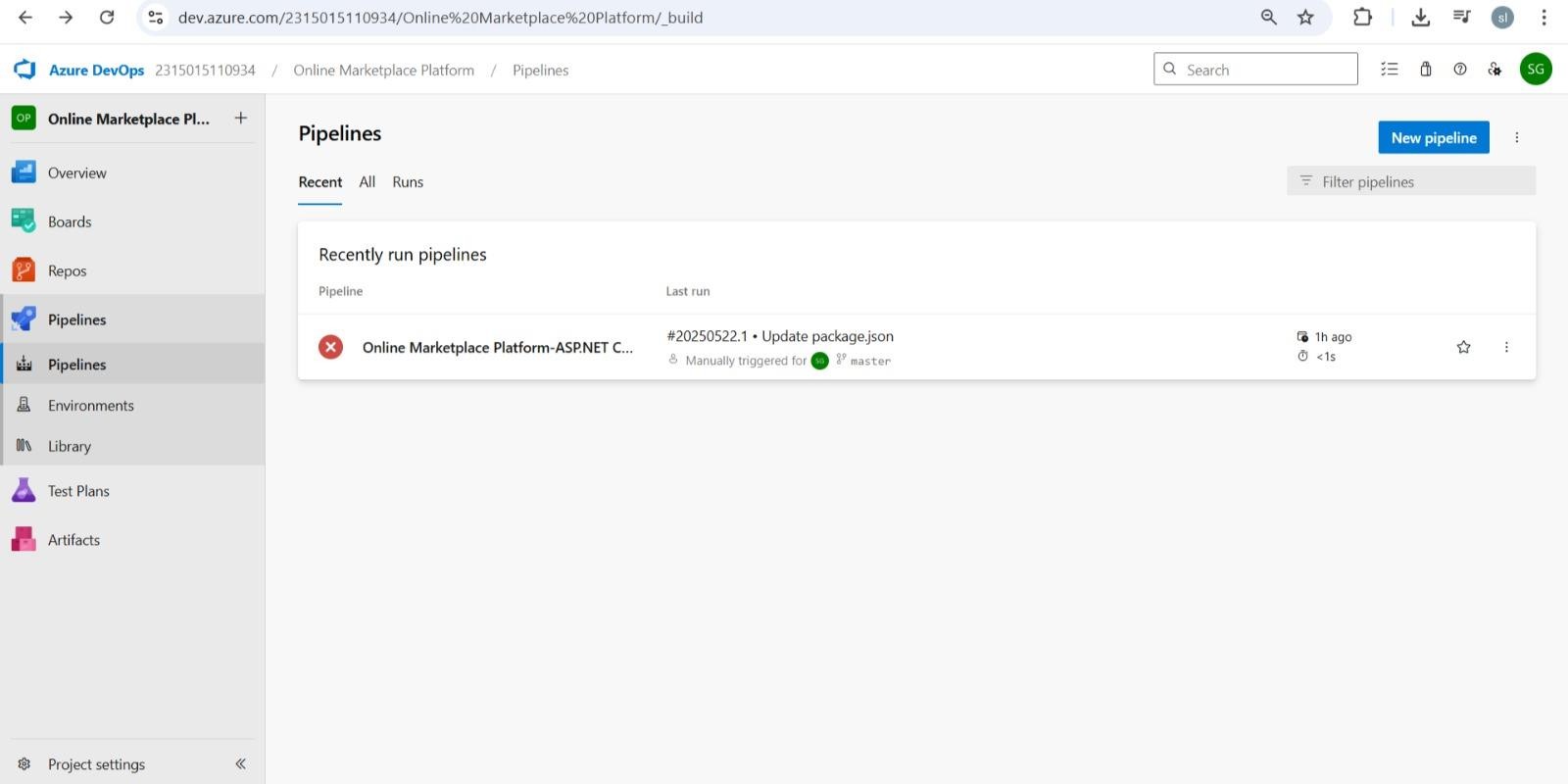
1. Configure the CD stage with deployment tasks such as deploying to Azure App Service, running database migrations or scripts, and restarting services using theAzureApp Service Deploy task linked to your subscription and app details.
2. SetTriggersandApprovals

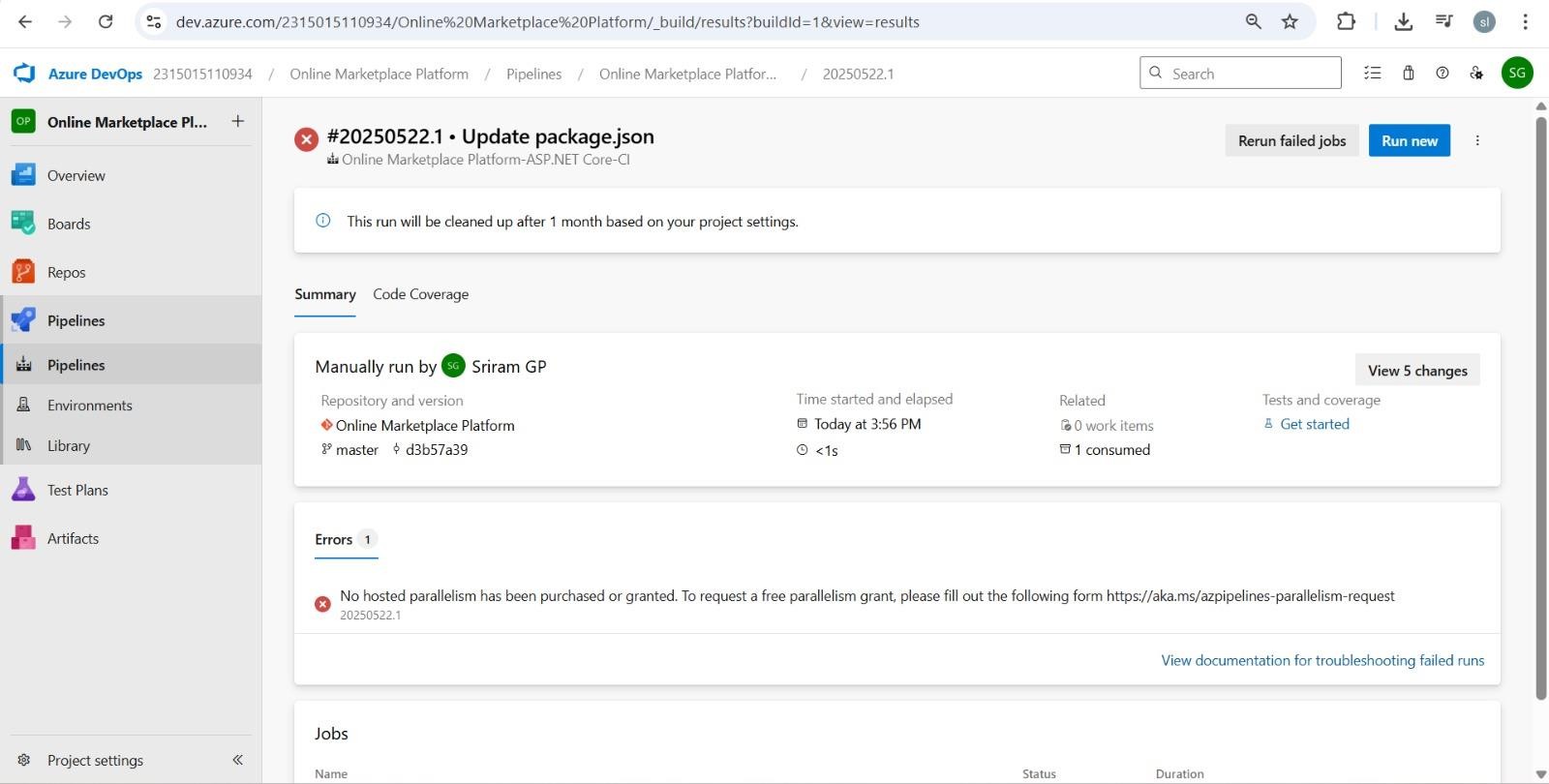
Enable continuous deployment trigger so the release pipeline runs automatically after a successful build. Forproductionenvironments, configurepre-deploymentapprovals to ensure manual verificationbefore release.

1. MonitorPipelinesandManageLogs ViewallpipelinerunsundertheRunssection. Checklogsforbuild/test/deploystagestodebuganyerrors.

YoucanalsointegrateemailalertsorMicrosoftTeamsnotificationsforbuildfailures.

1. ReviewandMaintain Pipelines RegularlyupdateyourpipelinetasksorYAMLconfigurationsasyourapplicationgrows. Ensure pipeline runs are clean and artifacts are stored securely. Integratequalitygatesandcodecoveragepoliciestomaintaincode quality.





## RESULT

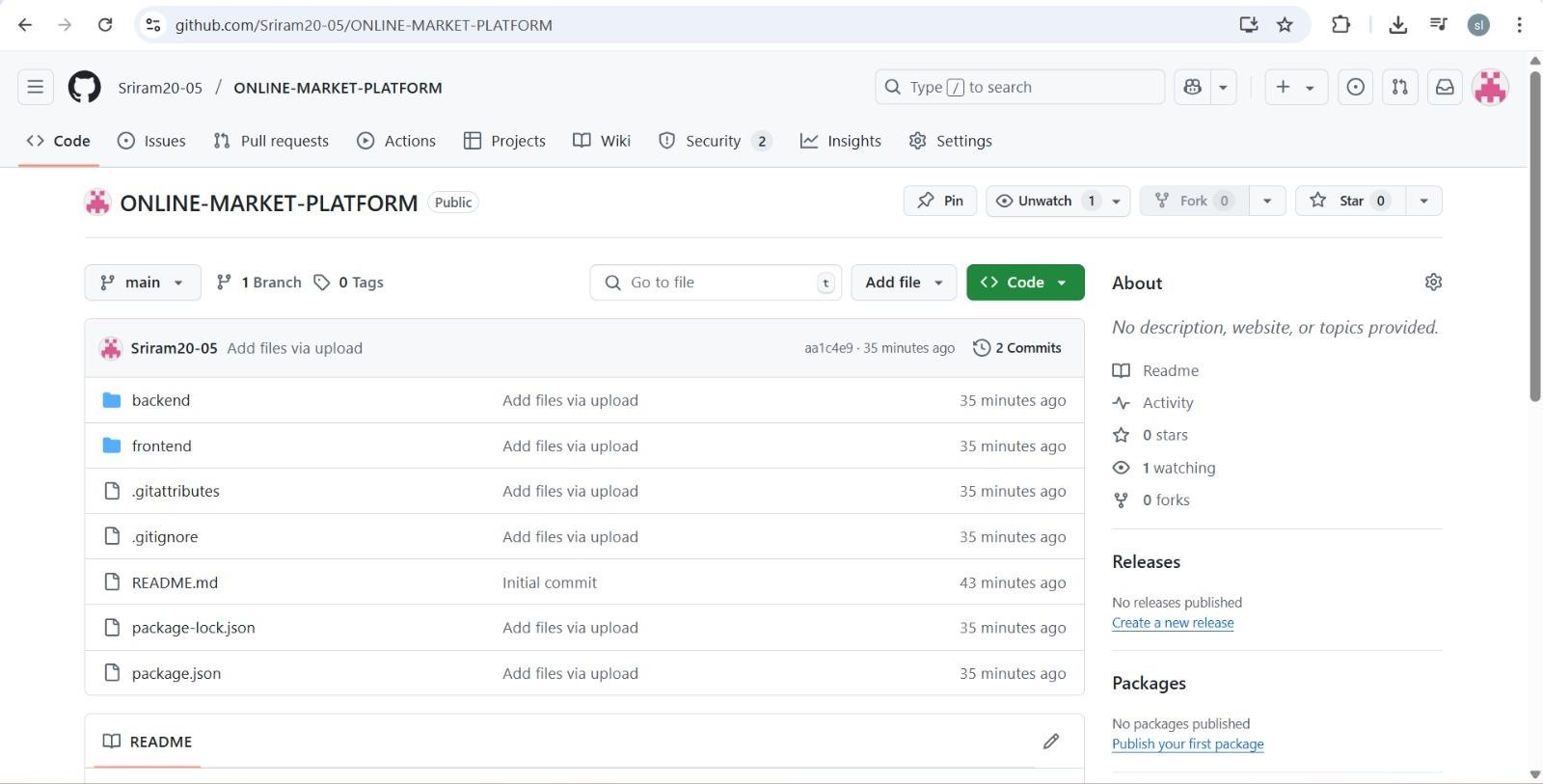
Thus,thepipelinesforthegivenproject **“Online Market Platform”**hasbeenexecuted successfully.

|  |  |
| --- | --- |
| **EXPNO:10** | **GITHUB:PROJECTSTRUCTURE&NAMING**  **CONVENTIONS** |

**Aim:**

Toprovideaclearandorganizedviewoftheproject'sfolderstructureandfilenamingconventions, helping contributors and users easily understand, navigate, and extend the Online market platform

**GitHubProjectStructure**

****

**Result:**

TheGitHubrepositoryclearlydisplaystheorganizedprojectstructureandconsistentnaming

conventions,makingiteasyforusersandcontributorstounderstandandnavigatethecodebase.