RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

DEPARTMENT OF INFORMATION TECHNOLOGY

LAB MANUAL

CS23432 – Software Construction

(REGULATION 2023)

RAJALAKSHMI ENGINEERING COLLEGE
Thandalam, Chennai-602015

Name:	Anushaa Maai T M
Register No:	231001015
Year / Branch / Section:	2nd /IT/ FA
Semester:	IV
Academic Year:	2024 - 2025

INDEX

Exp.No: 1	AZURE DEVOPS ENVIRONMENT SETUP
Date:22/01/2025	

AIM: To set up and access the Azure DevOps environment by creating an organization through the Azure Portal.

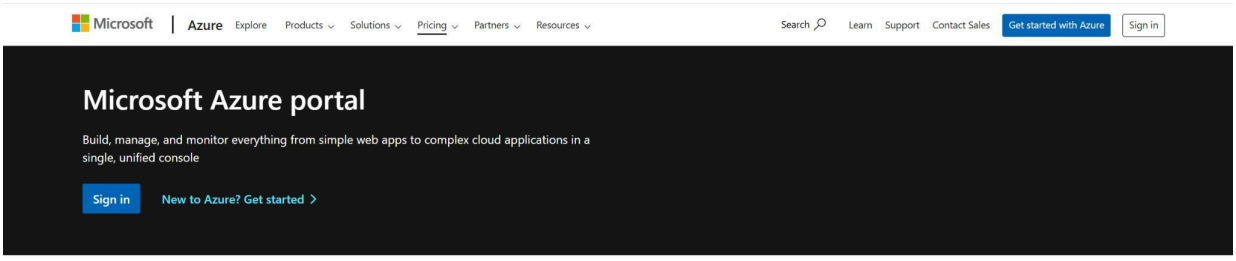
INSTALLATION

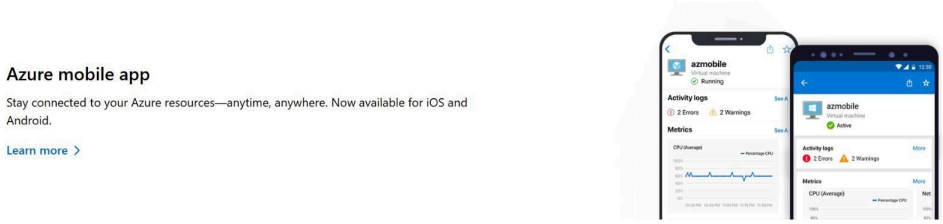
1. Open your web browser and go to the Azure website:

https://azure.microsoft.com/en-us/getstarted/azure-portal.

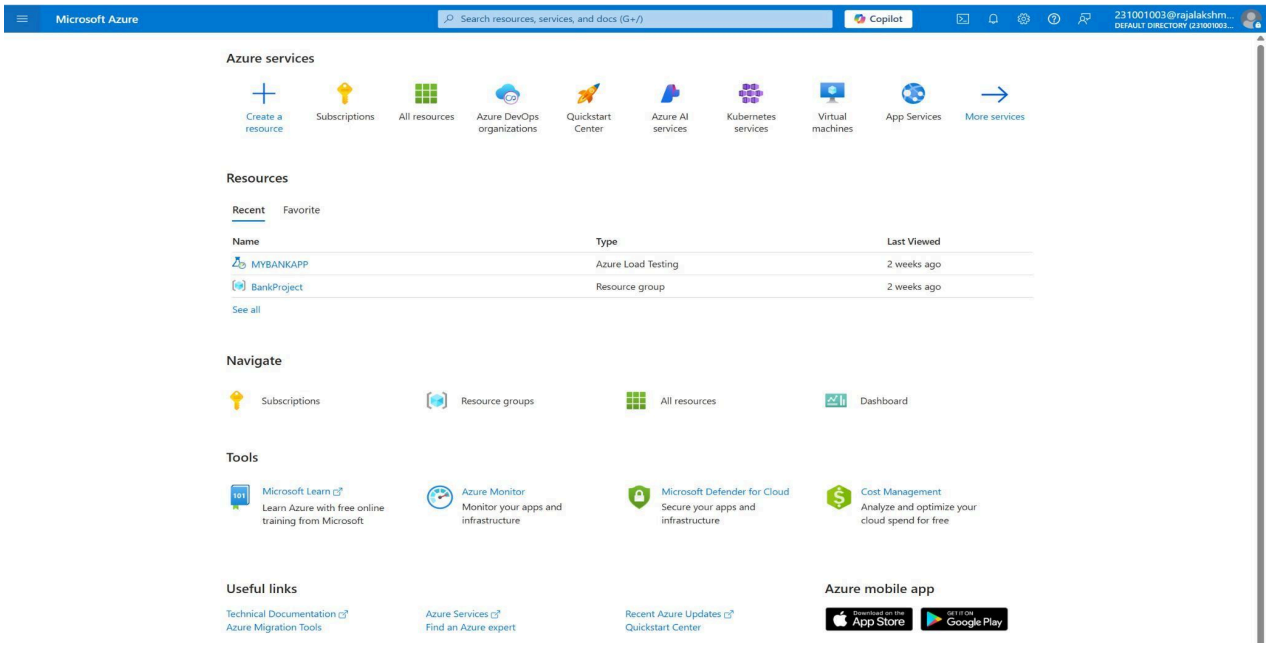
Sign in using your Microsoft account credentials.

If you don't have a Microsoft account, you can create one here: https://signup.live.com/?lic=1

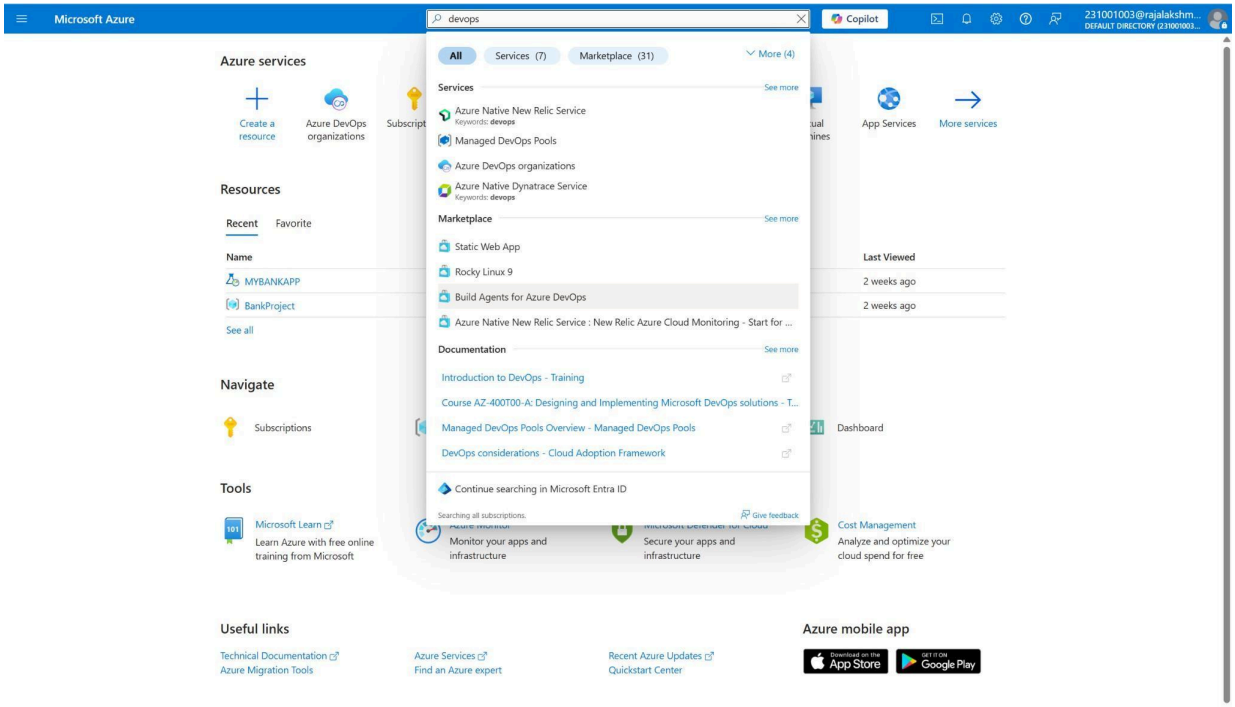




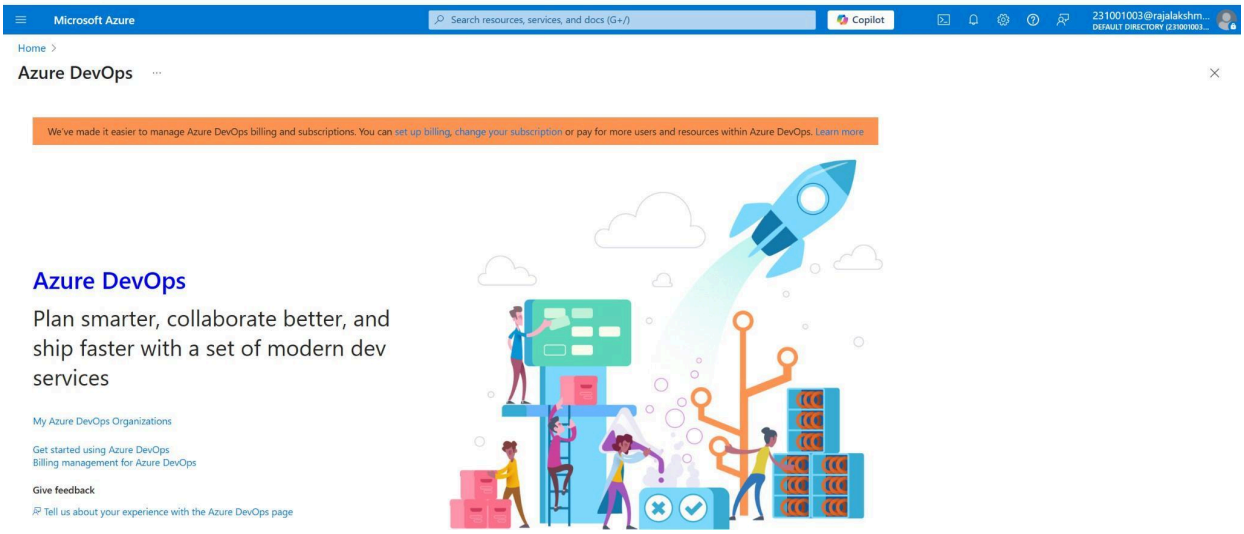
2. Azure home page



3. Open DevOps environment in the Azure platform by typing ***Azure DevOps Organizations*** in the search bar.



4. Click on the ***My Azure DevOps Organization*** link and create an organization and you should be taken to the Azure DevOps Organization Home page.



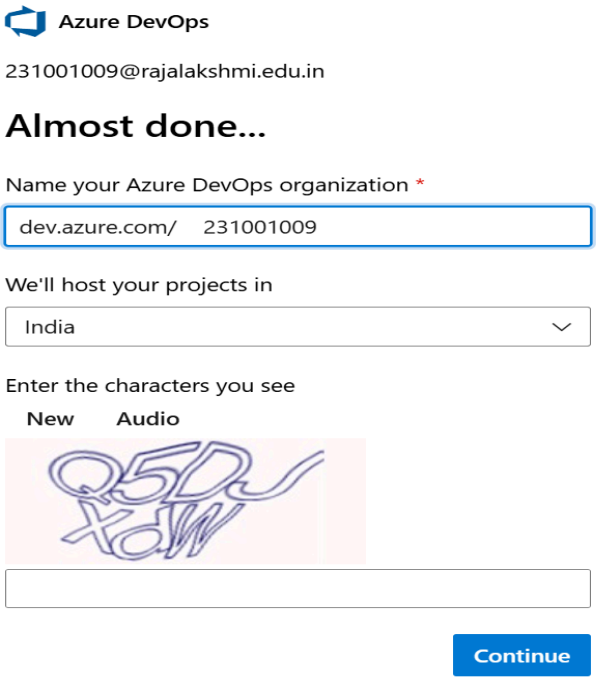
RESULT: Successfully accessed the Azure DevOps environment and created a new organization through the Azure portal.

Exp.No: 2	AZURE DEVOPS PROJECT SET UP AND USER STORY MANAGEMENT
Date: 07/02/2025	

AIM: To set up an Azure DevOps project for efficient collaboration and agile work management.

STEPS:

1. Create An Azure Account



2. Create the First Project in Your Organization

   a. After the organization is set up, you'll need to create your first **project**. This is where you'll begin to manage code, pipelines, work items, and more.

   b. On the organization's **Home page**, click on the **New Project** button.

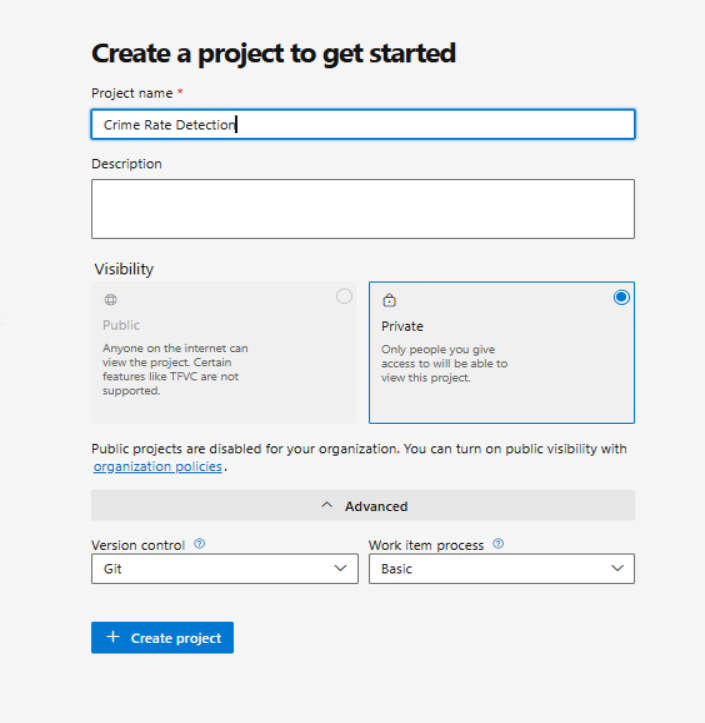
   c. Enter the project name, description, and visibility options:

      ***Name**:* Choose a name for the project (e.g., LMS).

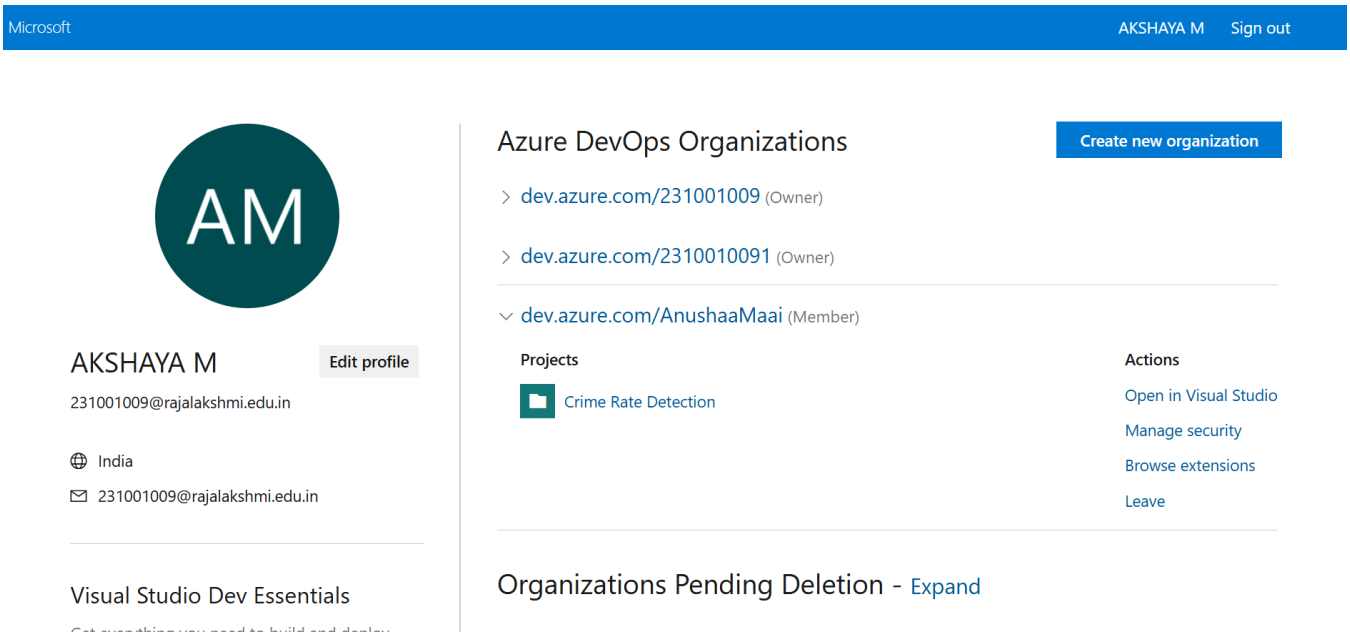
      ***Description**:* Optionally, add a description to provide more context about the project.

      ***Visibility**:* Choose whether you want the project to be **Private** (accessible only to those invited) or **Public** (accessible to anyone).

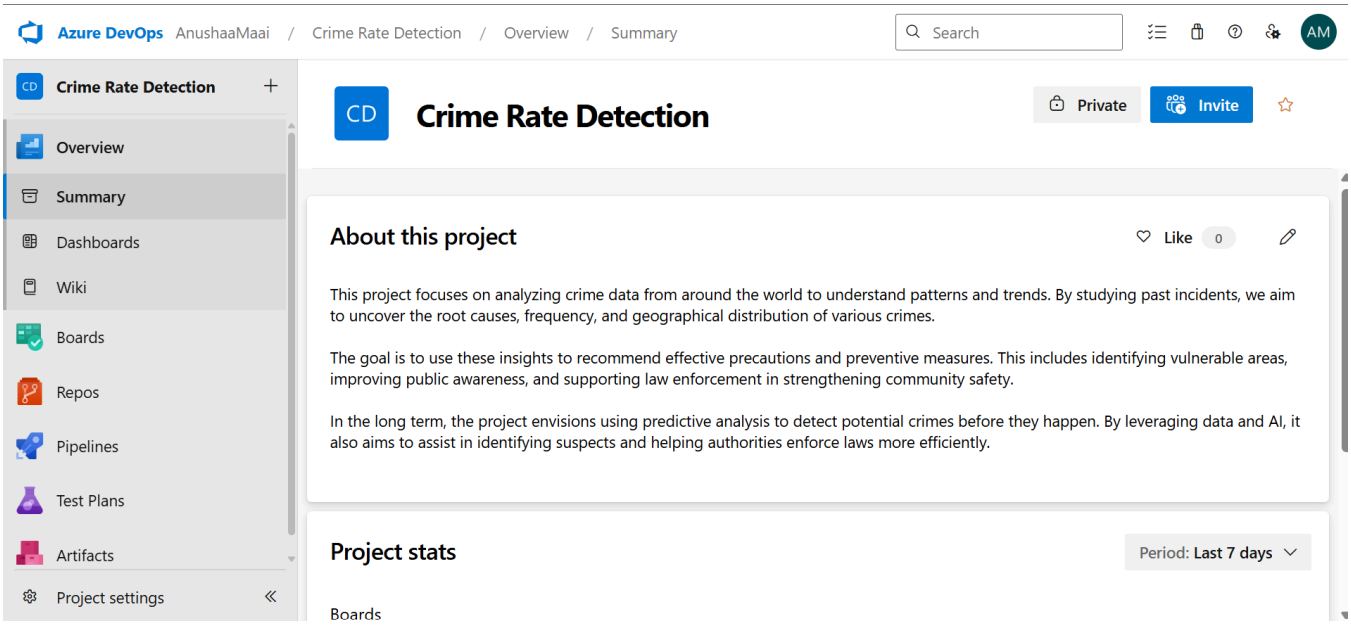
   d. Once you've filled out the details, click **Create** to set up your first project.



3. Once logged in, ensure you are in the correct organization. If you're part of multiple organizations, 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 the Azure DevOps Organization Home page.



4. Project dashboard



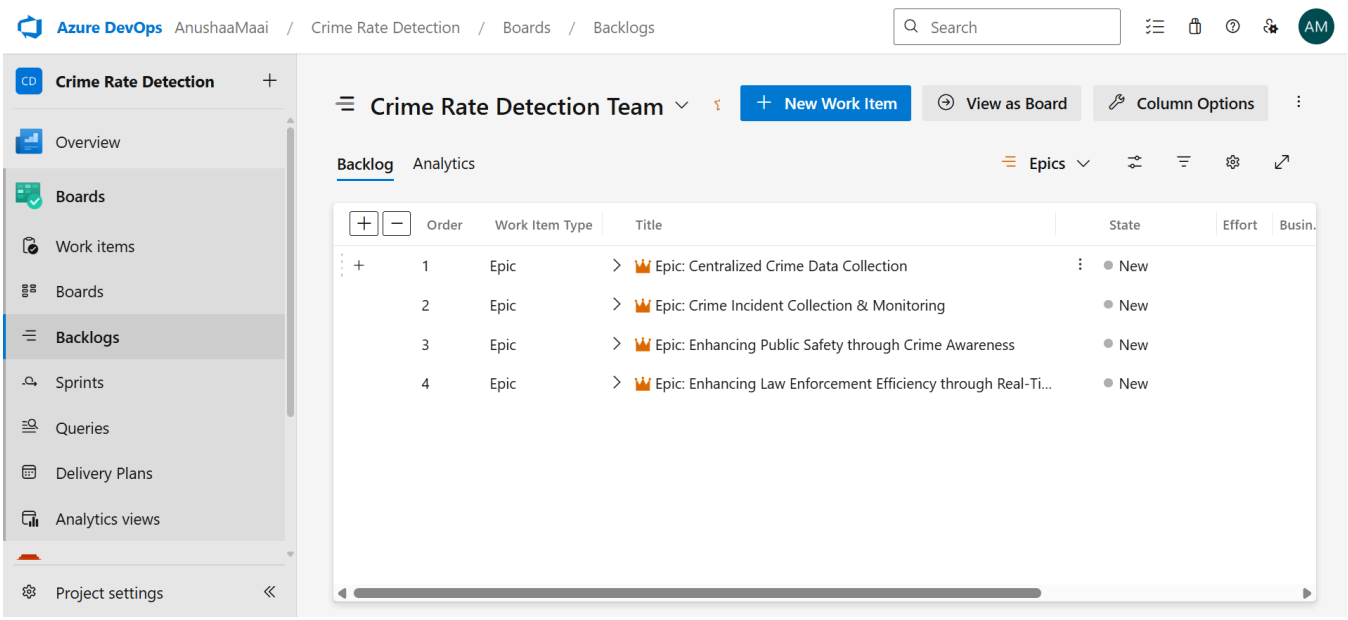
5. To manage user stories:

   a. From the **left-hand navigation menu**, click on **Boards**. This will take you to the main **Boards** page, where you can manage work items, backlogs, and sprints.

   b. On the **work items** page, you'll see the option to **Add a work item** at the top.

      Alternatively, you can find a + button or **Add New Work Item** depending on the view you're in.

      From the **Add a work item** dropdown, select **User Story**. This will open a form to enter details for this user story.

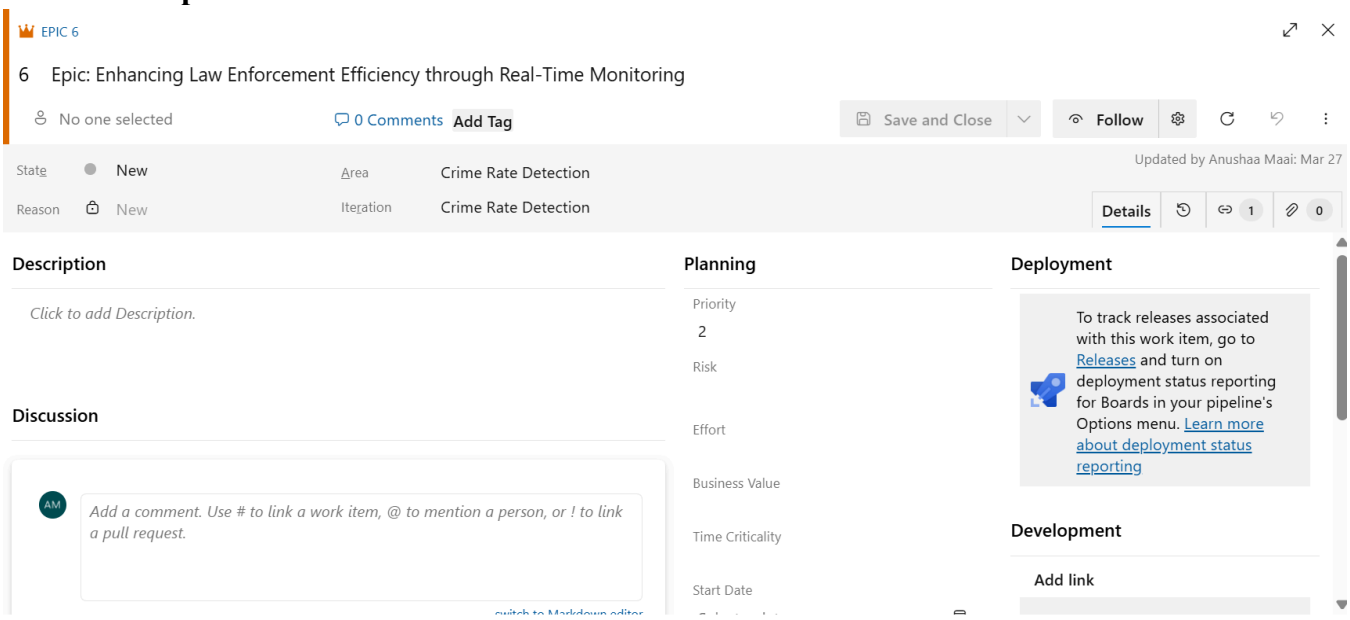


RESULT: Successfully created an Azure DevOps project with user story management and agile workflow setup.

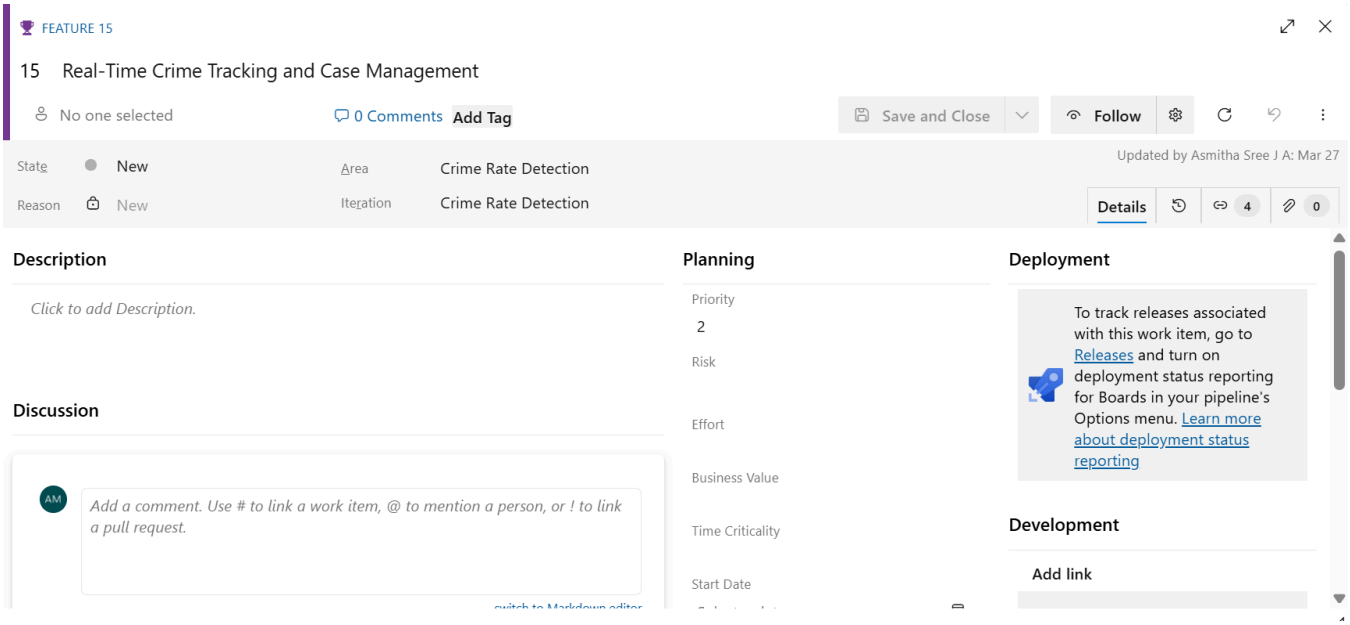
Exp.No: 3	SETTING UP EPICS FEATURES AND USER STORIES FOR PROJECT PLANNING
Date: 14/02/2025	

AIM: To learn about how to create epics, user story, features, backlogs for your assigned project.

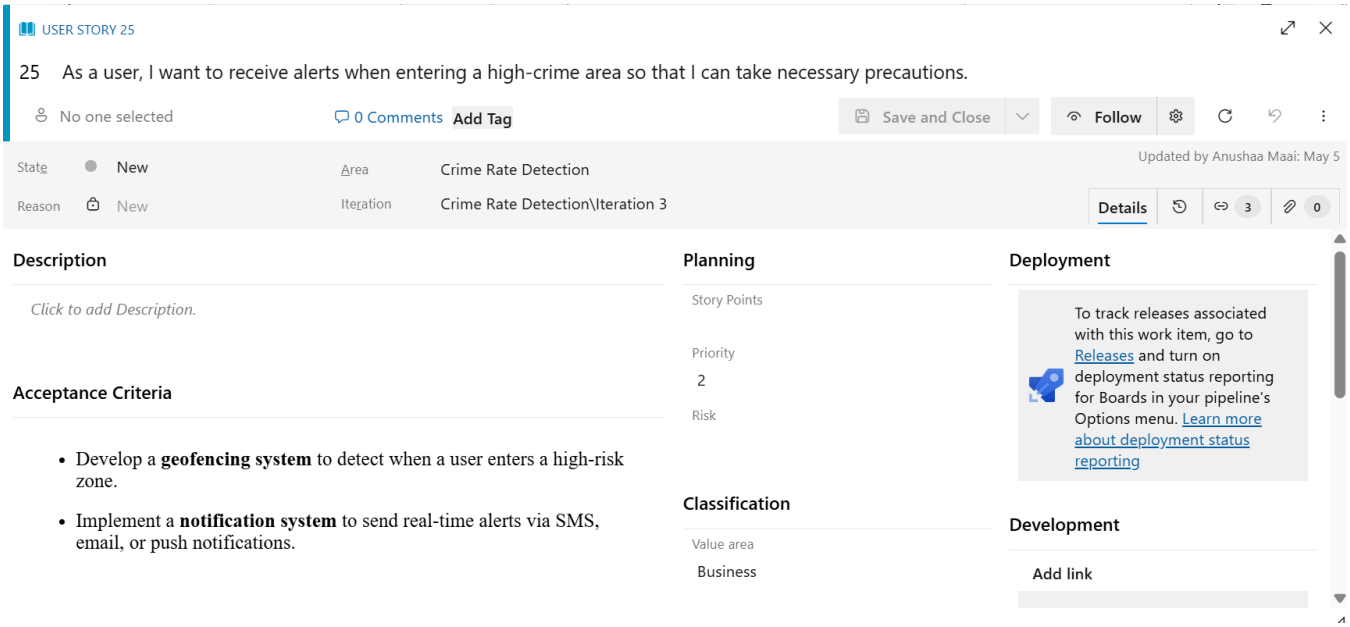
1. Fill in Epics



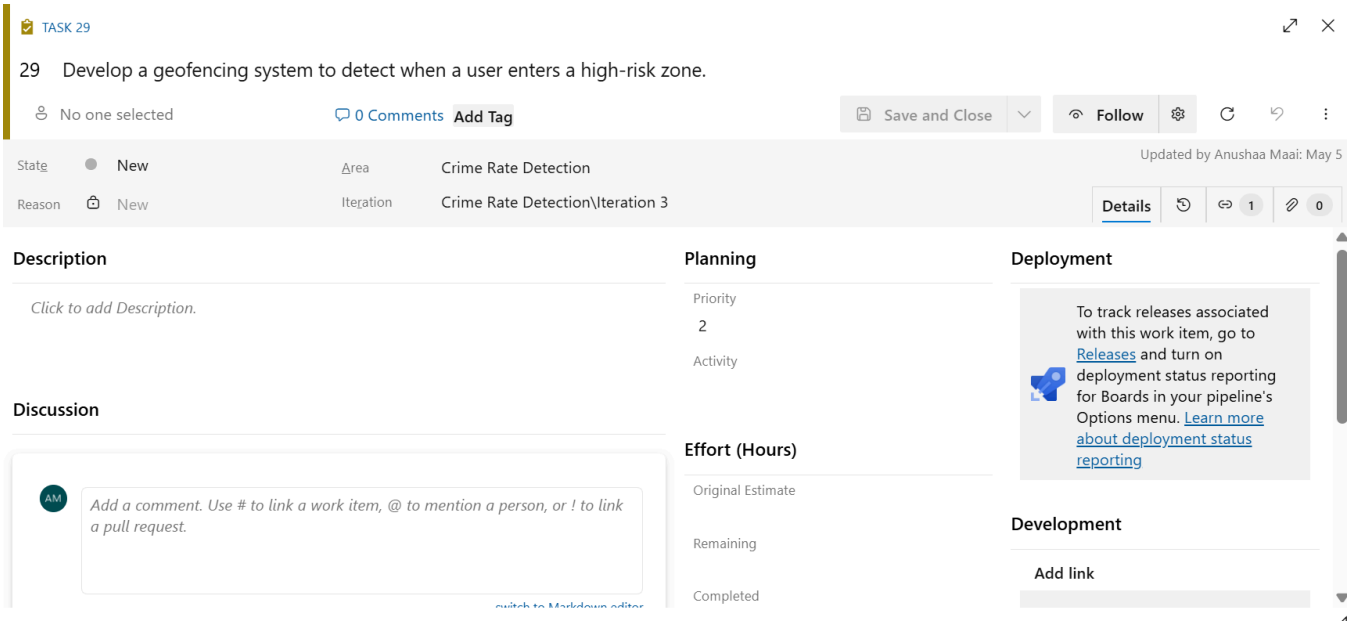
2. Fill in Features



3. Fill in User Story Details



4. Fill in Task Details



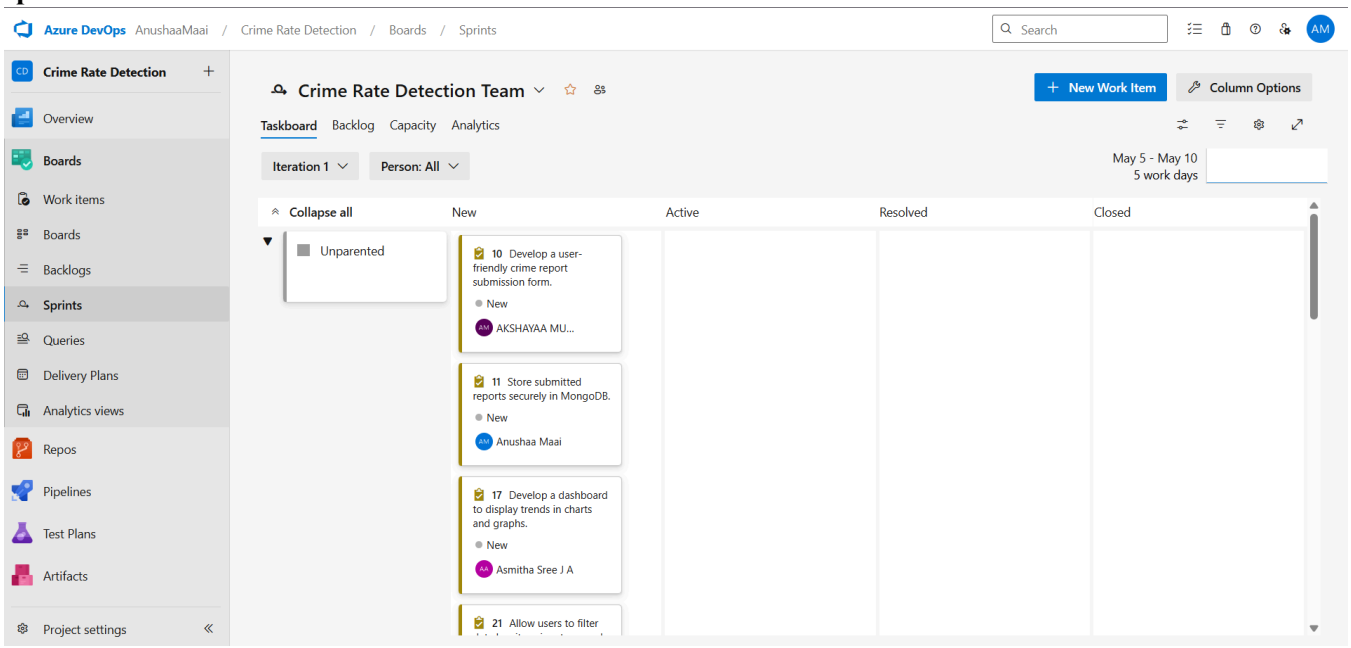
RESULT: Thus, the creation of epics, features, user story and task has been created successfully.

Exp.No: 4	SPRINT PLANNING
Date: 20/03/2025	

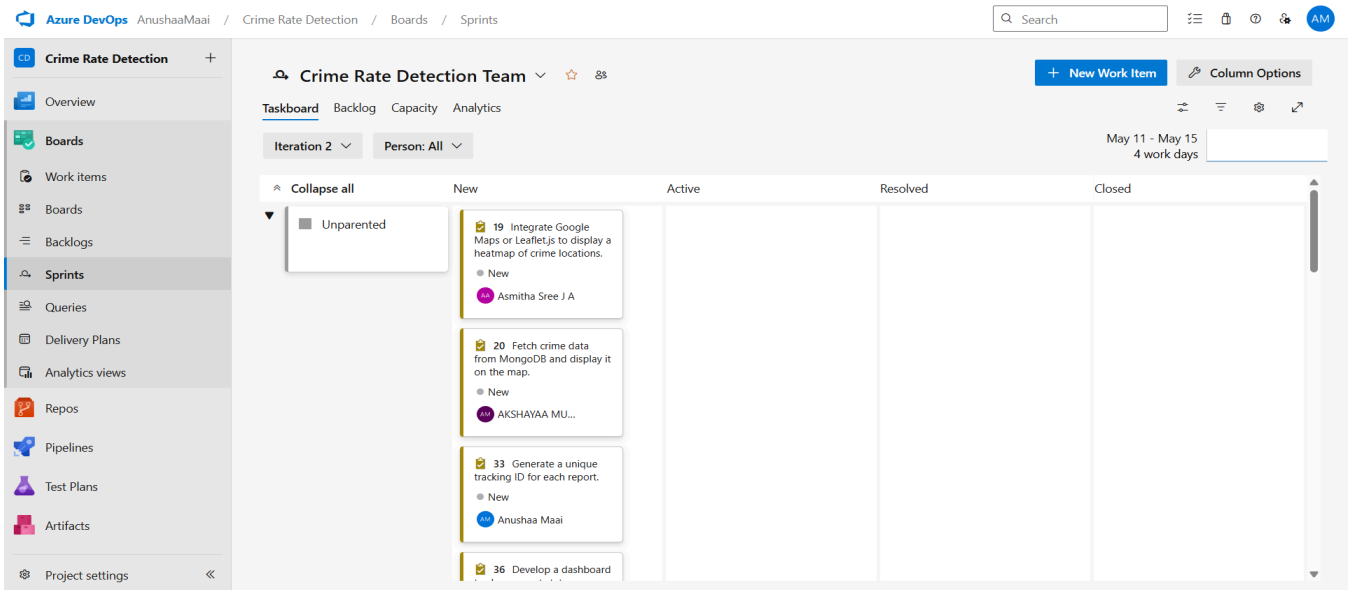
AIM: To assign user story to specific sprint for the Crime Rate Detection Project.

SPRINT PLANNING

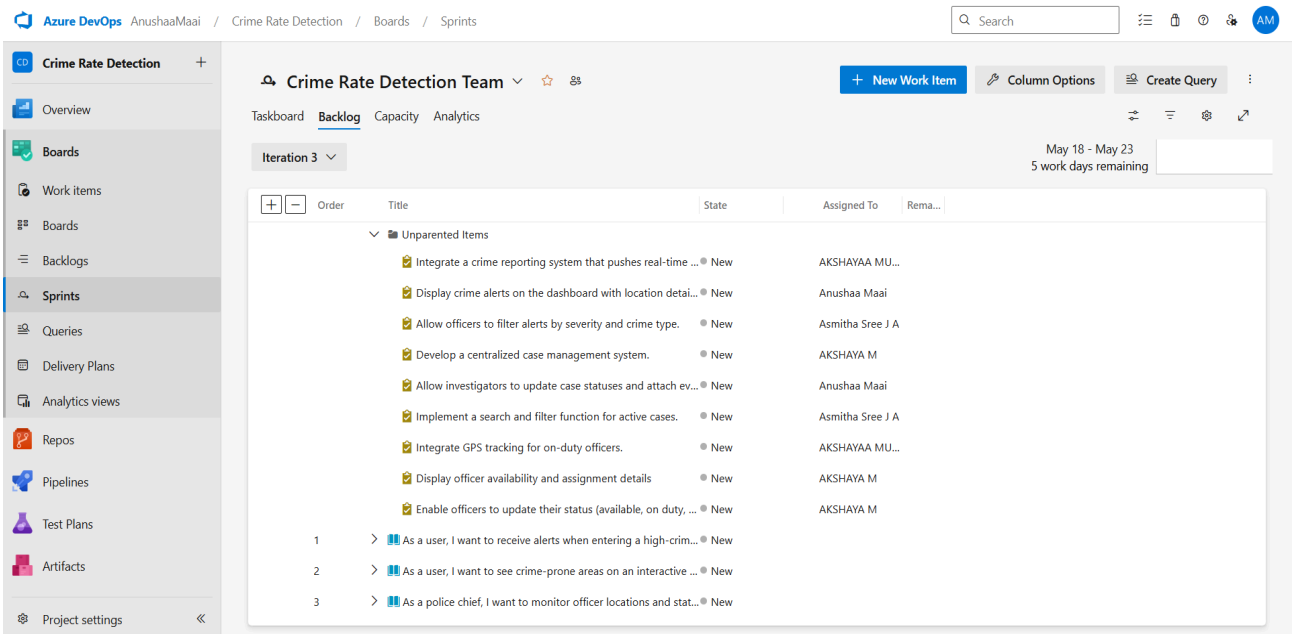
Sprint 1



Sprint 2



Sprint 3

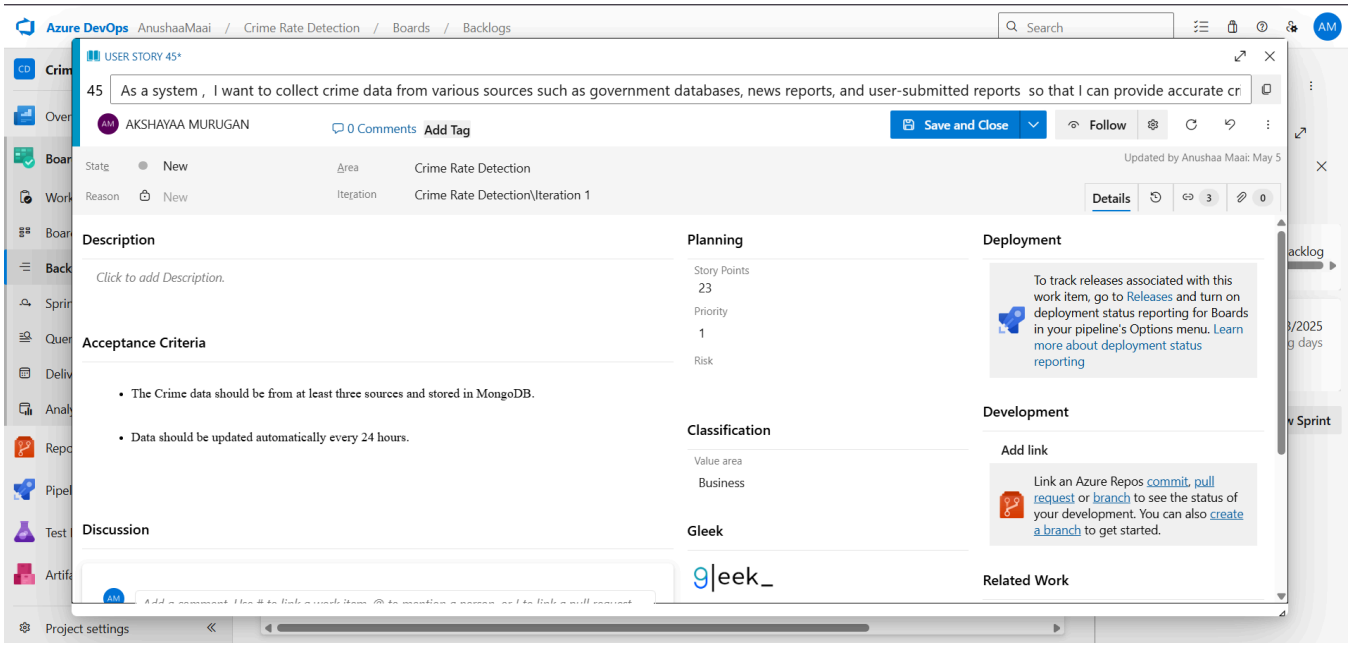


RESULT: The Sprints are created for the Crime Rate Detection Project.

Exp.No: 5	POKER ESTIMATION
Date: 28/03/2025	

AIM: Create Poker Estimation for the user stories - Crime Rate Detection Project.

POKER ESTIMATION

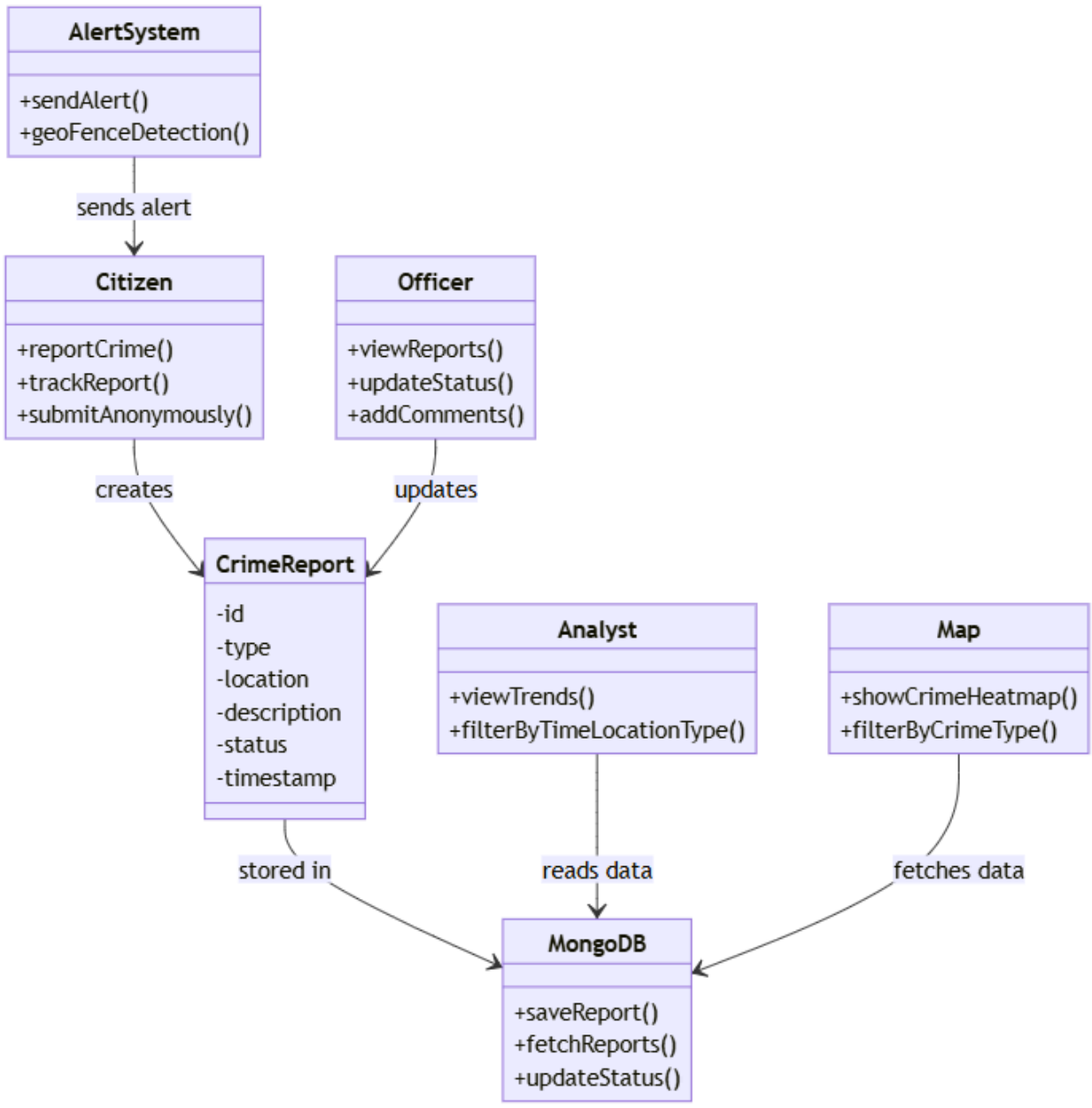


RESULT: The Estimation/Story Points is created for the project using Poker Estimation.

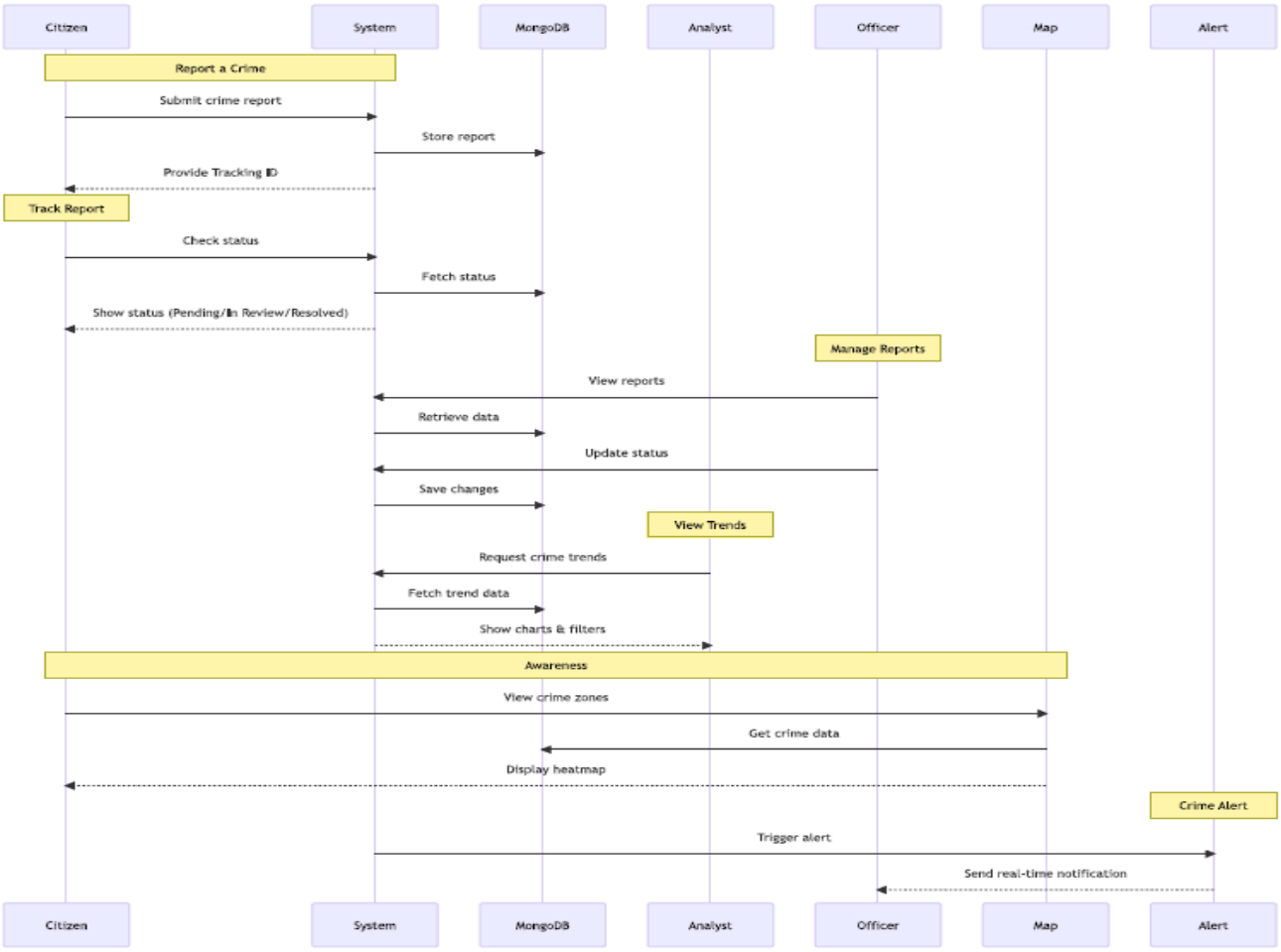
Exp.No: 6	DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE
Date: 04/04/2025	

AIM: To Design a Class Diagram and Sequence Diagram for the given Project.

6A. CLASS DIAGRAM



6B. SEQUENCE DIAGRAM

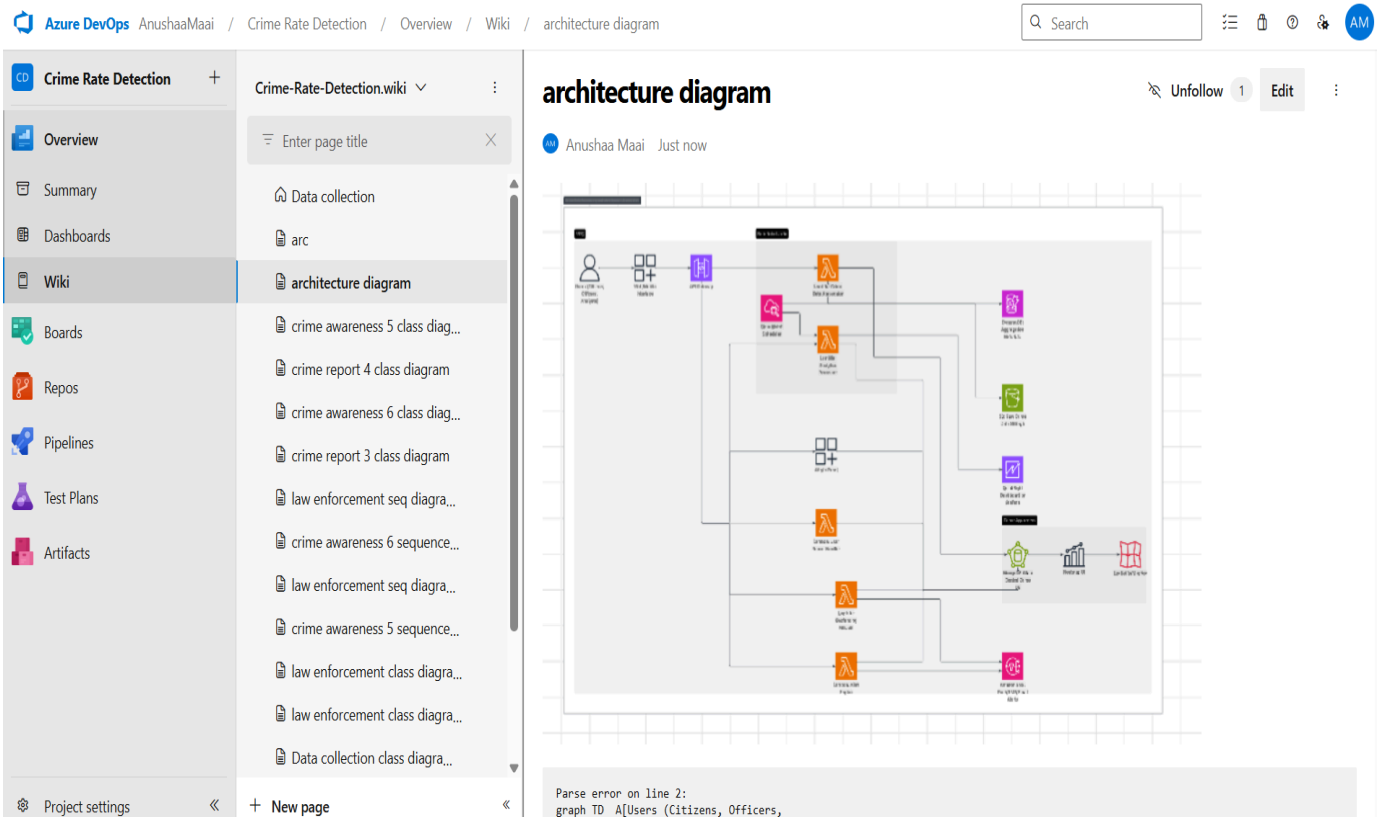


RESULT: The Class Diagram and Sequence Diagram is designed Successfully for the Crime Rate Detection Project.

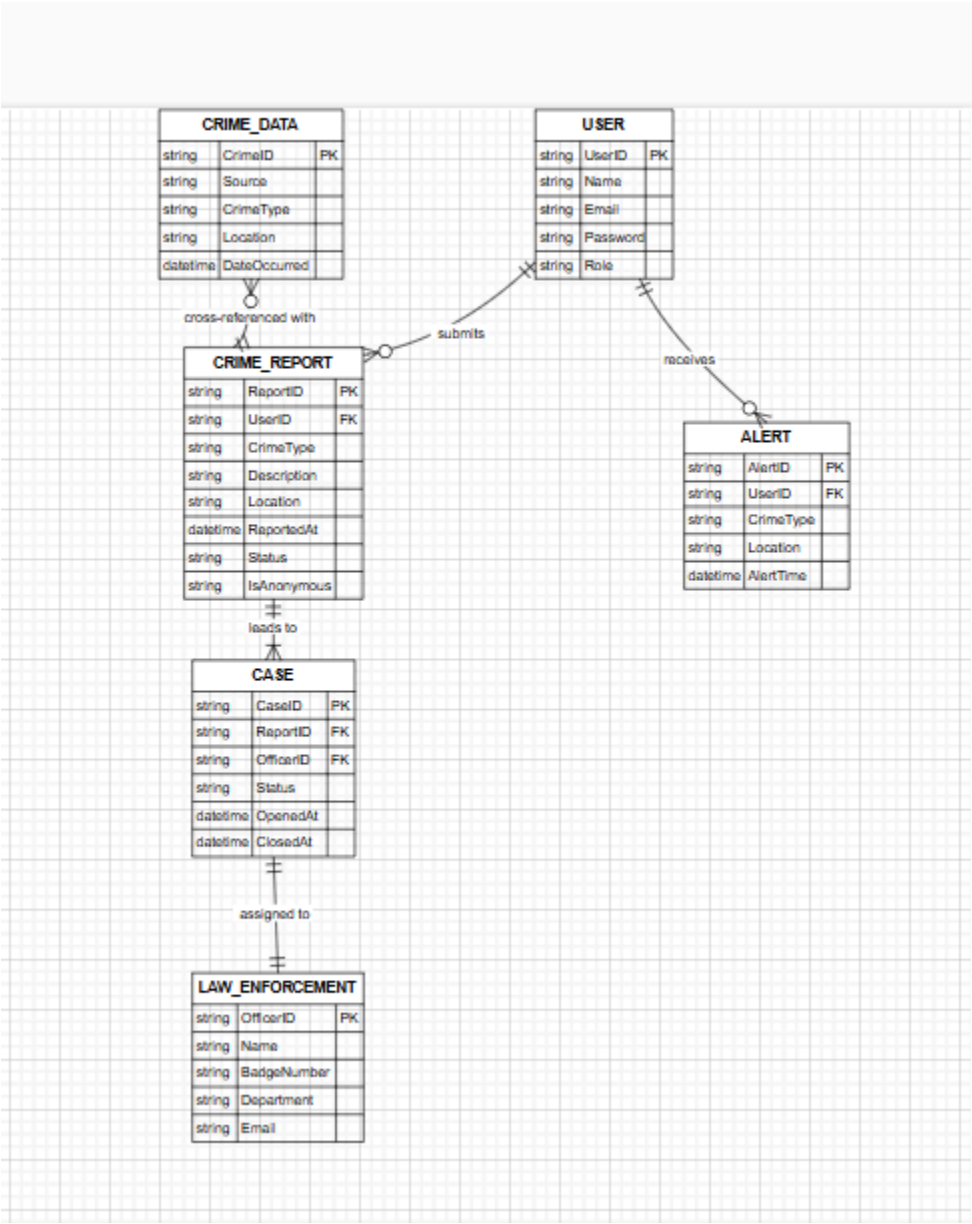
Exp.No: 7	DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE
Date: 16/04/2025	

AIM: To Design an Architectural Diagram and ER Diagram for the given Project.

7A. ARCHITECTURAL DIAGRAM



7B. ER DIAGRAM



RESULT: The Architecture Diagram and ER Diagram is designed Successfully for the Crime Rate Detection Project.

Exp.No: 8	TESTING-TEST PLANS AND TEST CASES
Date: 18/04/2025	

AIM: Test Plans and Test Case and write two test cases for at least five user stories showcasing the happy path and error scenarios in azure DevOps platform.

TEST PLANNING AND TEST CASE

Test Case Design Procedure

1. Understand Core Features of the Application

- User Authentication and Access Control
- Crime Data Collection and Aggregation
- Real-Time Crime Reporting and Monitoring
- Real-Time Crime Alerts
- Geofencing and Crime Heatmap Visualization
- Crime Analytics and Trend Reporting
- Suspect and Case Management
- Officer Deployment and Tracking

2. Define User Interactions
Each test case reflects realistic user actions:

- Submit Crime Report
- Track Report Status
- View Crime Map
- Receive Alerts
- Officer Reviews Reports
- System Fetches Data
- Admin Updates Dashboard

3. Design Happy Path Test Cases

- Focused on successful flows where inputs are valid and systems behave as expected.
- Examples:
    - Officer logs in with valid credentials
    - Receives reports successfully
    - Views crime map or crime details

4. Design Error Path Test Cases

- Focused on handling invalid, missing, or unexpected inputs.
- Examples:
    - Officer tries to log in with incorrect credentials.
    - Crime report submission fails due to missing location.
    - Tracking fails due to invalid tracking ID.

5. Break Down Steps and Expected

Results Each test case includes:

- Clear step-by-step actions
- Specific expected results for each step
  This ensures reproducibility and easy automation later.

6. Use Clear Naming and IDs

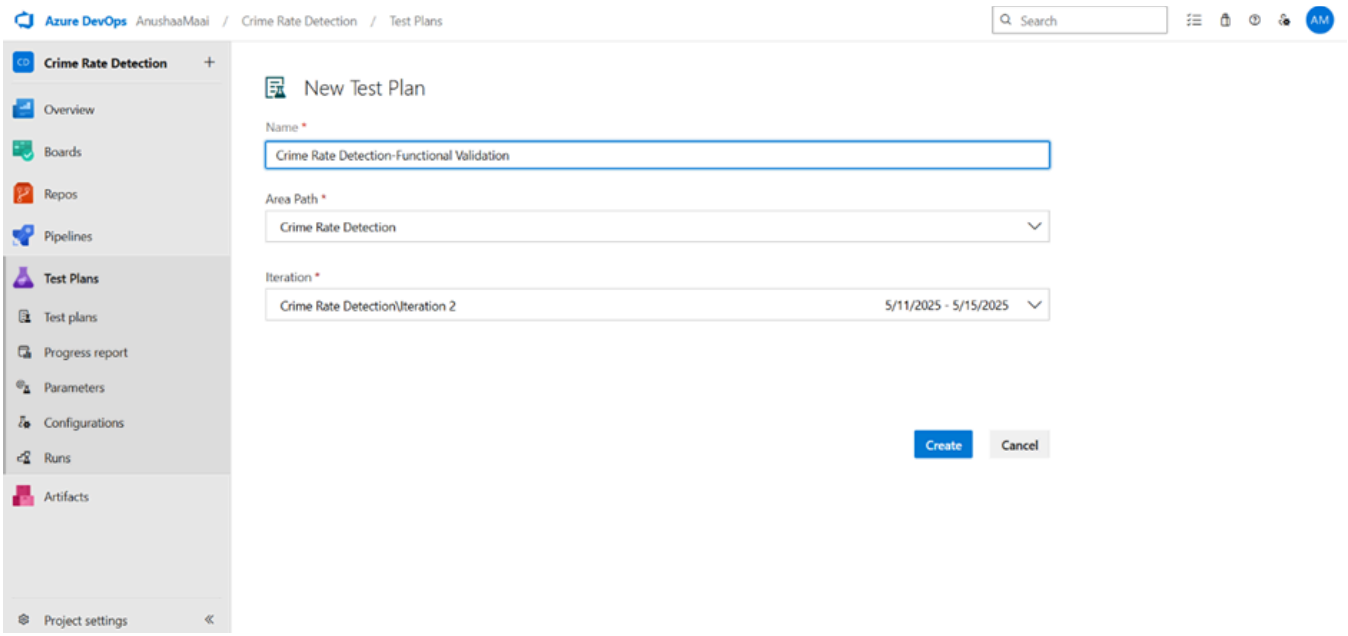
- Example:
    - TC85 – Collecting Crime Data from Multiple Sources
    - TC89 – Reporting crimes in the system
    - TC91 – Receiving Real-Time Crime Alerts
- Helps trace test cases to user stories and features in Azure DevOps

7. Separate Test Suites

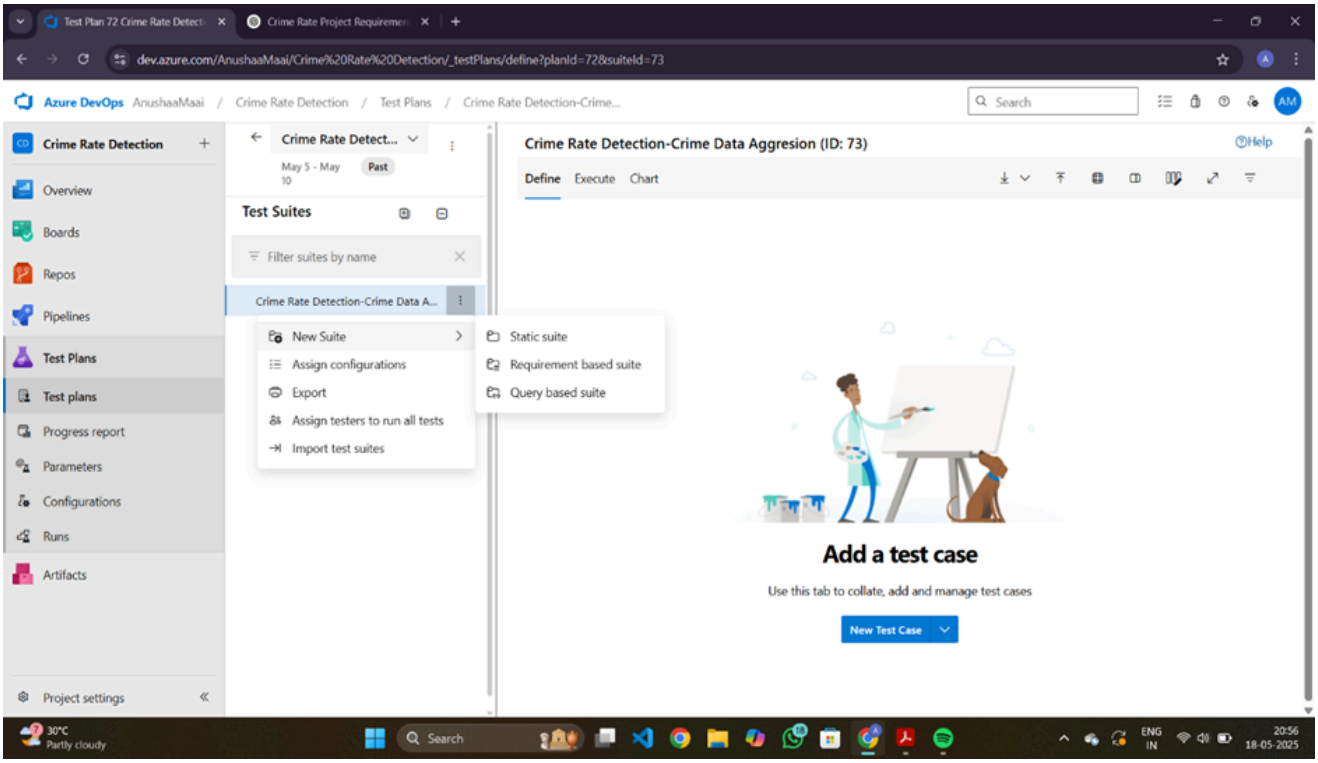
Organize test cases based on functional modules:

- TS01 – Crime Data Collection
- TS02 – Crime Alerts
- TS03 – Crime reporting
- TS04 – Crime Areas

1.New test plan



2. Test suite



3. Test case

Give two test cases for at least five user stories showcasing the happy path and error scenarios in azure DevOps platform.

Crime Rate Detection– Test Plans

User Stories – Crime Rate Detection

- **US45**: As a system , I want to collect crime data from various sources such as government databases, news reports, and user-submitted reports so that I can provide accurate crime insights.
- **US60**: As a police officer, I want to receive real-time alerts on reported crimes so that I can respond quickly and efficiently.
- **US51**:As a citizen, I want to report a crime through a secure platform so that authorities can take appropriate action.
- **US66**: As a user, I want to see crime-prone areas on an interactive map so that I can stay informed and avoid dangerous locations.

TEST SUITE: TS01 – CRIME DATA COLLECTION (ID: 84)

Test Case: TC01 – Collecting crime data from multiple sources

User Story: As a system , I want to collect crime data from various sources such as government databases, news reports, and user-submitted reports so that I can provide accurate crime insights.

Test Type: Happy Path

Steps and Expected Results:

- **Action:** Trigger the scheduled data collection job.
  Expected Result: System attempts to connect to all sources.
- **Action:** One data source (e.g., news API) is unavailable.
  Expected Result: System logs an error for that source and skips it.
- **Action:** Continue fetching data from remaining available sources.
  Expected Result: Data from available sources is stored successfully; a warning is logged for the skipped source.

Notes:

This test simulates partial failure where not all sources are reachable. The system should still process available data and alert for any failed connections without crashing.

Test Case: TC02 – Handling Data Source Failure

User Story: As a system , I want to collect crime data from various sources such as government databases, news reports, and user-submitted reports so that I can provide accurate crime insights.

Test Type: Error Path

Steps and Expected Results:

- **Action:** Trigger the data collection process (manual or scheduled)
  Expected Result: The system begins fetching data from all configured sources

- **Action:** Simulate one source failure (e.g., government API returns 500 error)
  Expected Result: System logs the failure but continues fetching from the other sources

- **Action:** Complete the process and store data
  Expected Result: Data from the successful sources is stored in MongoDB; an alert is logged for the failed one

Notes:

Verifies system resilience when an external data source becomes temporarily unavailable.

TEST SUITE: TS02 – CRIME ALERTS (ID: 86)

Test Case: TC01 – Receiving real-time crime alerts

User Story: As a police officer, I want to receive real-time alerts on reported crimes so that I can respond quickly and efficiently.

Test Type: Happy Path

Steps and Expected Results:

- **Action:** A new crime is reported through the system (user form or external source)
  Expected Result: The crime data is received and processed successfully by the backend

- **Action:** Real-time alert is generated and pushed to the officer's dashboard or mobile device
  Expected Result: Officer receives a notification with crime type, location, and timestamp

- **Action:** Officer clicks on the alert to view full details
  Expected Result: Full report opens with actionable information like suspect details, severity, and map view

Test Case: TC02 – Filtering Fails due to backend timeout

User Story: As a police officer, I want to receive real-time alerts on reported crimes so that I can respond quickly and efficiently.

Test Type: Error Path

Steps and Expected Results:

- **Action:** Open the real-time alert dashboard and apply a filter (e.g., "Severity = High")
  Expected Result: System attempts to filter alerts

- **Action:** Backend service experiences a timeout while processing filter request
  Expected Result: An error message appears (e.g., “Unable to filter alerts, please try again later”)

- **Action:** Retry after some time
  Expected Result: Filtered results load correctly when backend recovers

TEST SUITE: TS03 – CRIME REPORTING (ID: 89)

Test Case: TC01 – Reporting crimes in the system

User Story: As a citizen, I want to report a crime through a secure platform so that authorities can take appropriate action.

Test Type: Happy Path

Steps and Expected Results:

- **Action:** Navigate to the "Report Crime" page

  Expected Result: A secure form with input fields (crime type, description, location, etc.) is displayed

- **Action:** Fill in valid details and click "Submit"

  Expected Result: The report is submitted successfully and confirmation message is shown

- **Action:** Check backend or admin dashboard for new report entry

  Expected Result: The new crime report appears in the system with a unique ID and timestamp

Test Case: TC02 – Reporting Crime with Missing Required Fields

User Story:As a citizen, I want to report a crime through a secure platform so that authorities can take appropriate action.

Test Type: Error Path

Steps and Expected Results:

- **Action:** Navigate to the "Report Crime" page

  Expected Result: The crime reporting form is displayed

- **Action:** Leave required fields (e.g., description or location) empty and click "Submit"

  Expected Result: Form validation fails; error message is shown (e.g., "Location is required")

- **Action:** Try to proceed without correcting the input

  Expected Result: Submission is blocked until all required fields are properly filled

TEST SUITE: TS04 – CRIME AREAS (ID: 92)

Test Case: TC01 – Viewing Crime-Prone Areas on an Interactive Map

User Story: As a user, I want to see crime-prone areas on an interactive map so that I can stay informed and avoid dangerous locations.

Test Type: Happy Path

Steps and Expected Results:

- **Action:** Navigate to the crime map section of the application

  Expected Result: An interactive map is displayed with default view centered on user's city or region

- **Action:** Zoom or pan across different areas on the map

  Expected Result: Crime data dynamically updates and highlights high-risk zones

- **Action:** Click on a highlighted (crime-prone) area

  Expected Result: A popup or sidebar appears showing recent crimes, types, and timestamps for that location

Test Case: TC02 – Map Fails to Load Due to API Error

User Story: As a user, I want to see crime-prone areas on an interactive map so that I can stay informed and avoid dangerous locations.

Test Type: Error Path

Steps and Expected Results:

- **Action:** Navigate to the crime map section of the application

  Expected Result: The map fails to load due to a backend/API error

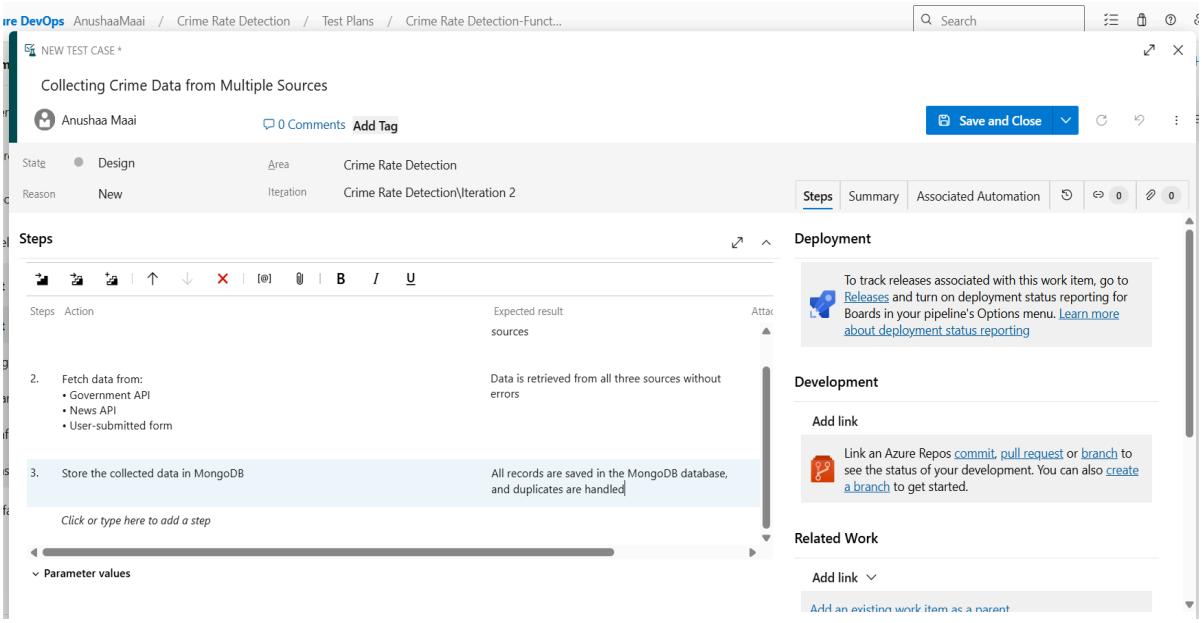
- **Action:** System attempts to retrieve map data from the crime data API

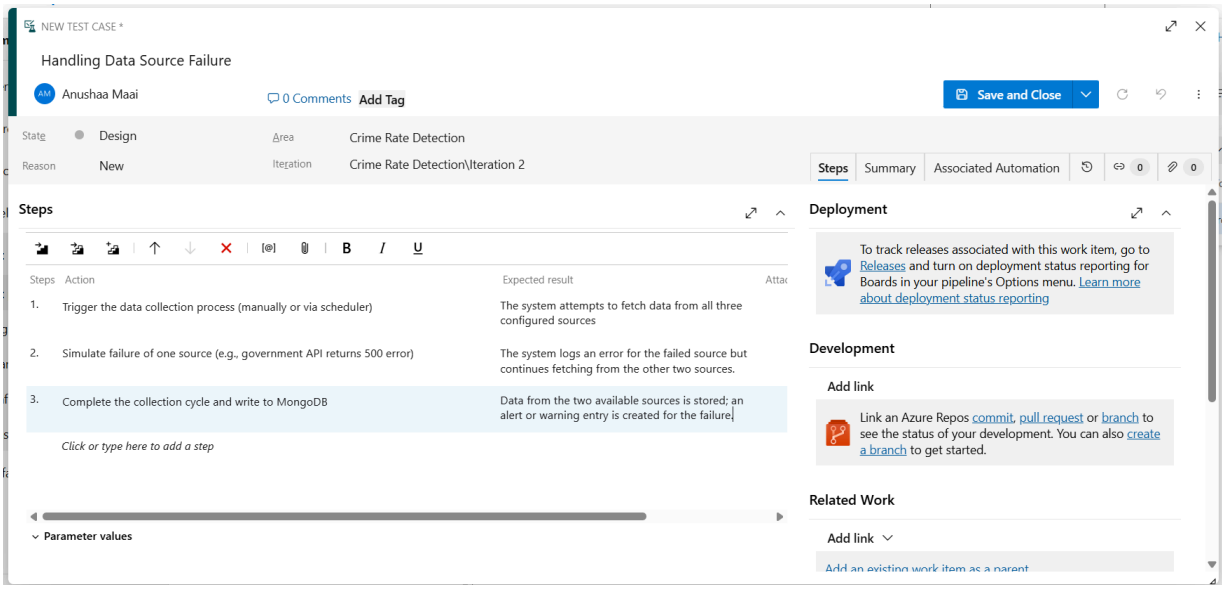
  Expected Result: API returns error (e.g., 500 Internal Server Error), and map cannot display data

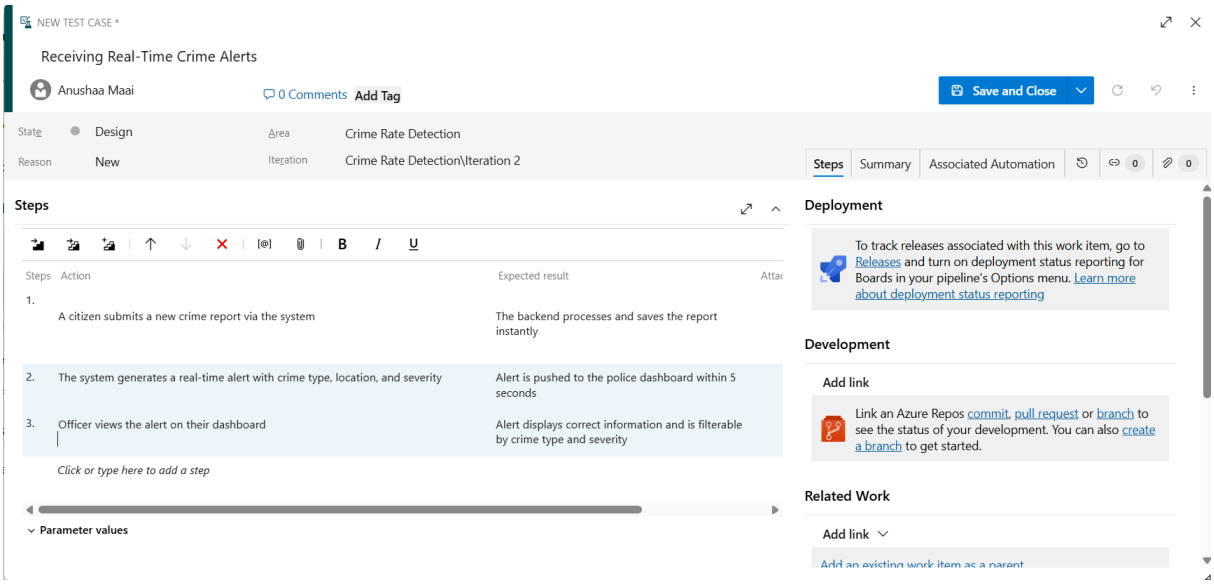
- **Action:** System displays error fallback UI

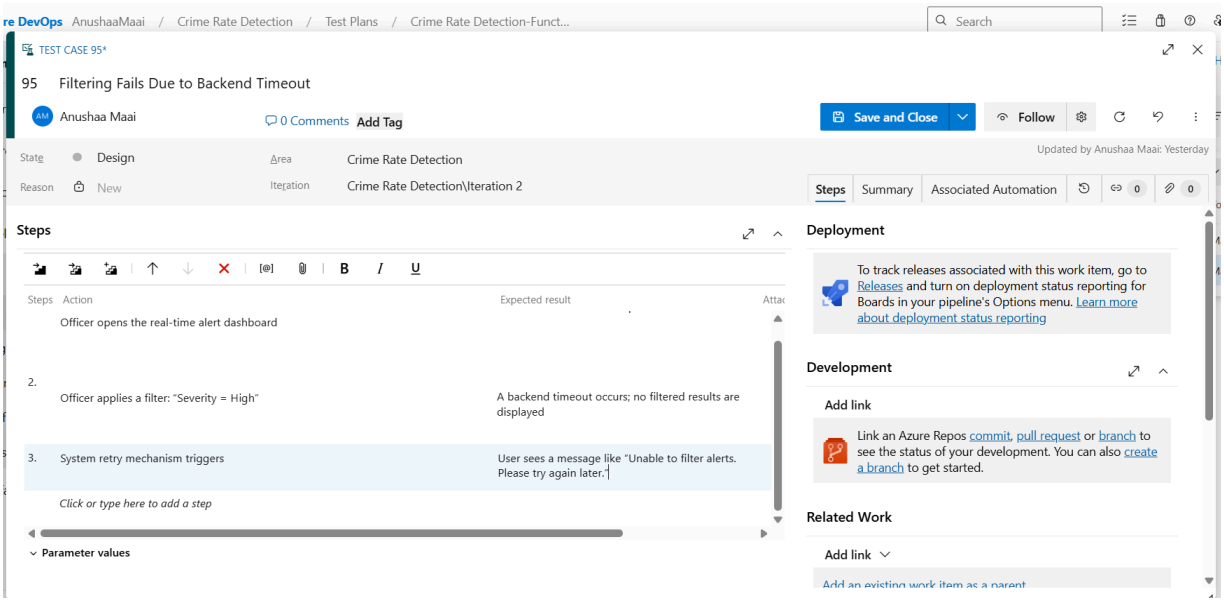
  Expected Result: User sees a message like “Unable to load map data. Please try again later.”

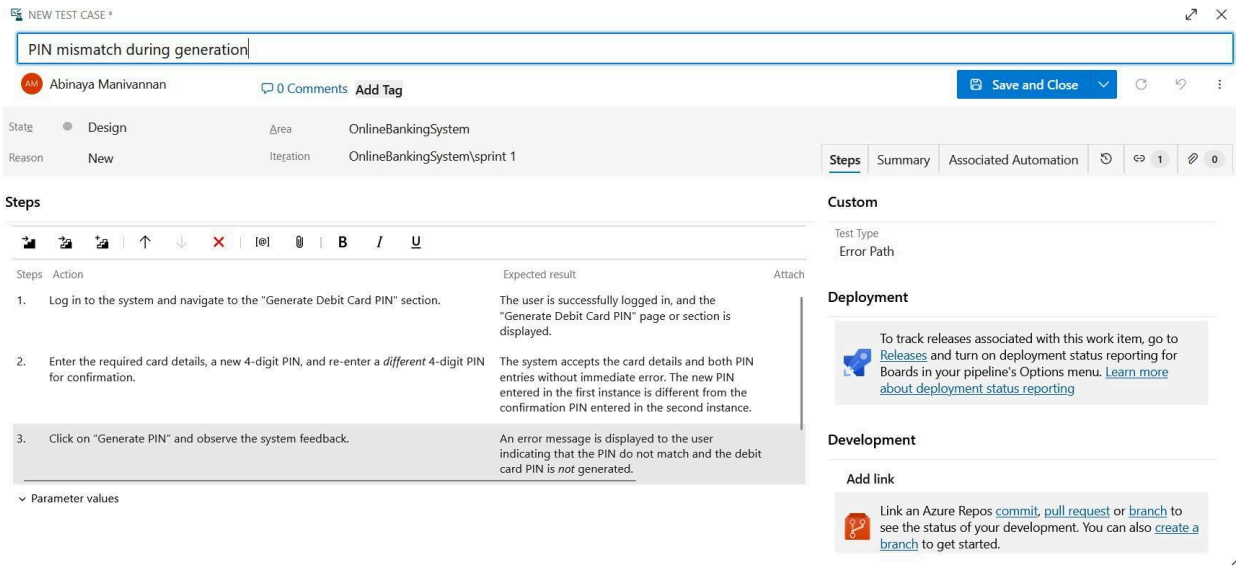
Test Cases


NEW TEST CASE *
Collecting Crime Data from Multiple Sources
Anushaa Maai
0 Comments
Add Tag
Save and Close
State
Design
Area
Crime Rate Detection
Reason
New
Iteration
Crime Rate Detection\Iteration 2
Steps
Summary
Associated Automation
Steps
Steps Action
Expected result
sources
2. Fetch data from:
• Government API
• News API
• User-submitted form
Data is retrieved from all three sources without errors
3. Store the collected data in MongoDB
All records are saved in the MongoDB database, and duplicates are handled
Click or type here to add a step
Parameter values
Deployment
To track releases associated with this work item, go to Releases and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting
Development
Add link
Link an Azure Repos commit, pull request or branch to see the status of your development. You can also create a branch to get started.
Related Work
Add link


NEW TEST CASE *
Handling Data Source Failure
Anushaa Maai
0 Comments
Add Tag
Save and Close
State
Design
Area
Crime Rate Detection
Reason
New
Iteration
Crime Rate Detection\Iteration 2
Steps
Summary
Associated Automation
Steps
Steps Action
Expected result
1. Trigger the data collection process (manually or via scheduler)
The system attempts to fetch data from all three configured sources
2. Simulate failure of one source (e.g., government API returns 500 error)
The system logs an error for the failed source but continues fetching from the other two sources.
3. Complete the collection cycle and write to MongoDB
Data from the two available sources is stored; an alert or warning entry is created for the failure
Click or type here to add a step
Parameter values
Deployment
To track releases associated with this work item, go to Releases and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting
Development
Add link
Link an Azure Repos commit, pull request or branch to see the status of your development. You can also create a branch to get started.
Related Work
Add link

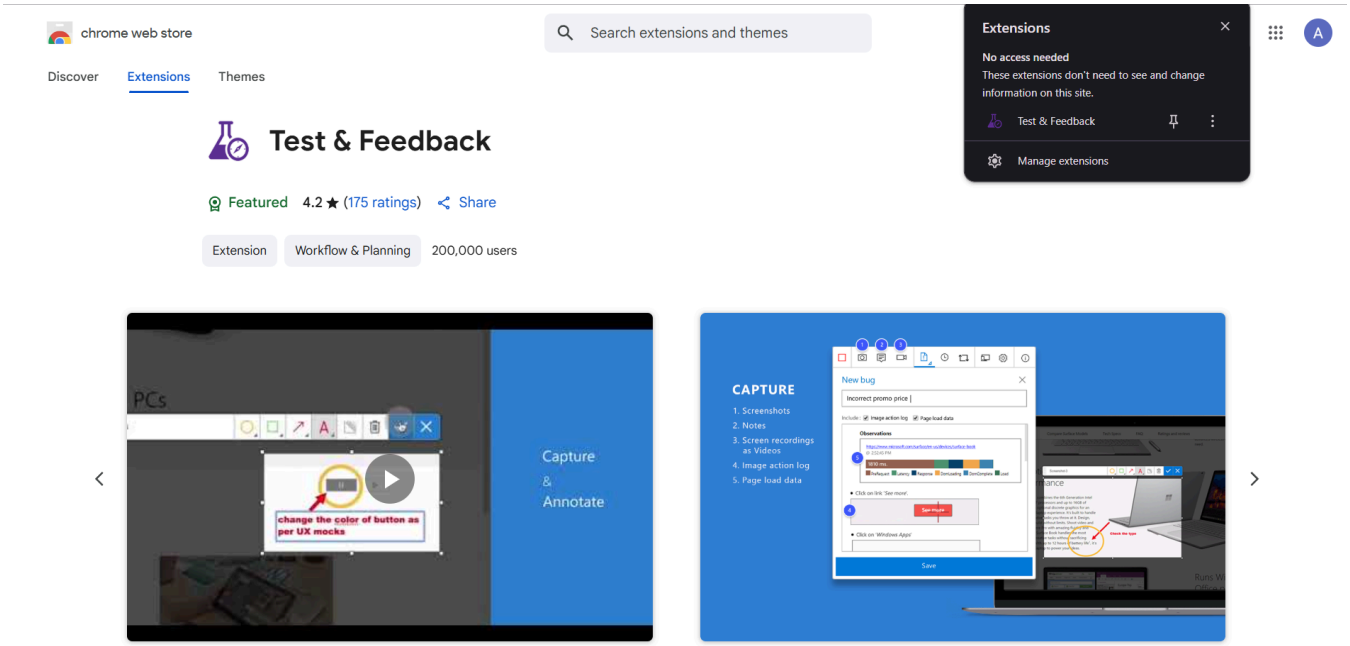

NEW TEST CASE *
Receiving Real-Time Crime Alerts
Anushaa Maai
0 Comments
Add Tag
Save and Close
State
Design
Area
Crime Rate Detection
Reason
New
Iteration
Crime Rate Detection\Iteration 2
Steps
Summary
Associated Automation
Steps
Steps
Action
Expected result
1.
A citizen submits a new crime report via the system
The backend processes and saves the report instantly
2.
The system generates a real-time alert with crime type, location, and severity
Alert is pushed to the police dashboard within 5 seconds
3.
Officer views the alert on their dashboard
Alert displays correct information and is filterable by crime type and severity
Click or type here to add a step
Parameter values
Deployment
To track releases associated with this work item, go to Releases and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting
Development
Add link
Link an Azure Repos commit, pull request or branch to see the status of your development. You can also create a branch to get started.
Related Work
Add link


re DevOps
AnushaaMaai
Crime Rate Detection
Test Plans
Crime Rate Detection-Funct...
Search
TEST CASE 95*
95
Filtering Fails Due to Backend Timeout
Anushaa Maai
0 Comments
Add Tag
Save and Close
Follow
Updated by Anushaa Maai: Yesterday
State
Design
Area
Crime Rate Detection
Reason
New
Iteration
Crime Rate Detection\Iteration 2
Steps
Summary
Associated Automation
Steps
Steps
Action
Expected result
Officer opens the real-time alert dashboard
2.
Officer applies a filter: "Severity = High"
A backend timeout occurs; no filtered results are displayed
3.
System retry mechanism triggers
User sees a message like "Unable to filter alerts. Please try again later."
Click or type here to add a step
Parameter values
Deployment
To track releases associated with this work item, go to Releases and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting
Development
Add link
Link an Azure Repos commit, pull request or branch to see the status of your development. You can also create a branch to get started.
Related Work
Add link

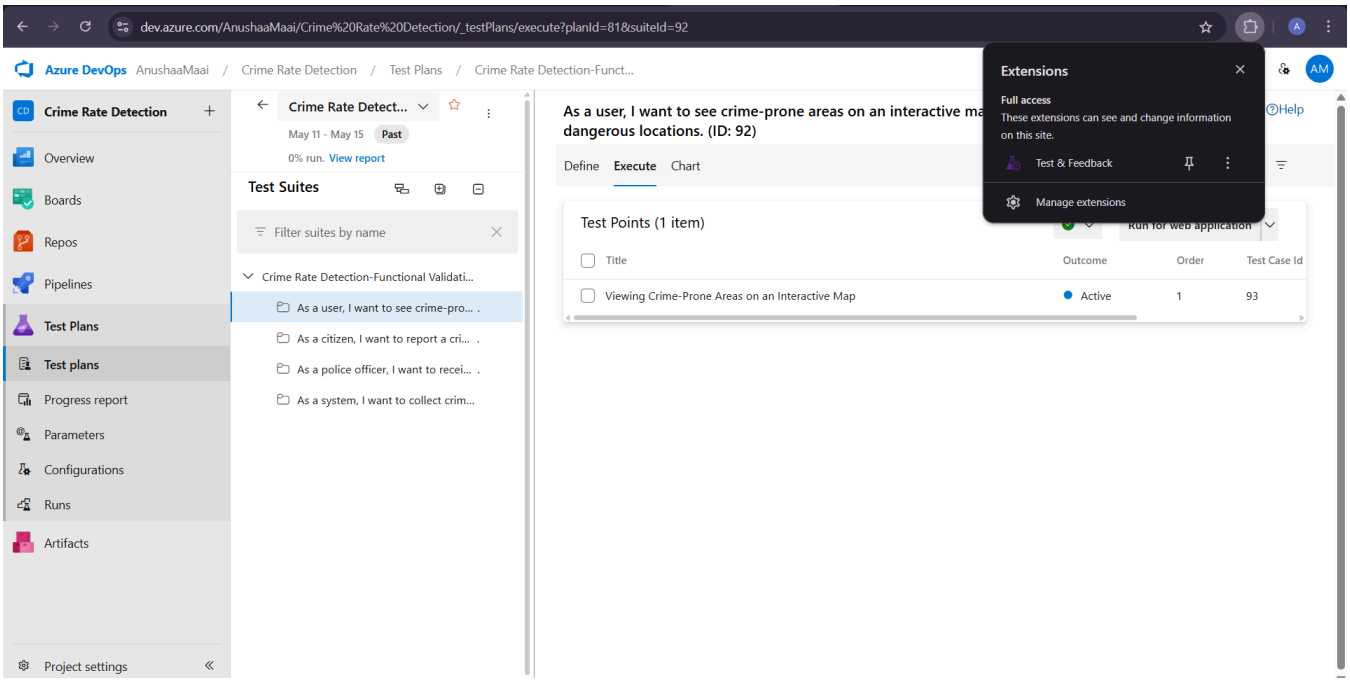


4. Installation of test

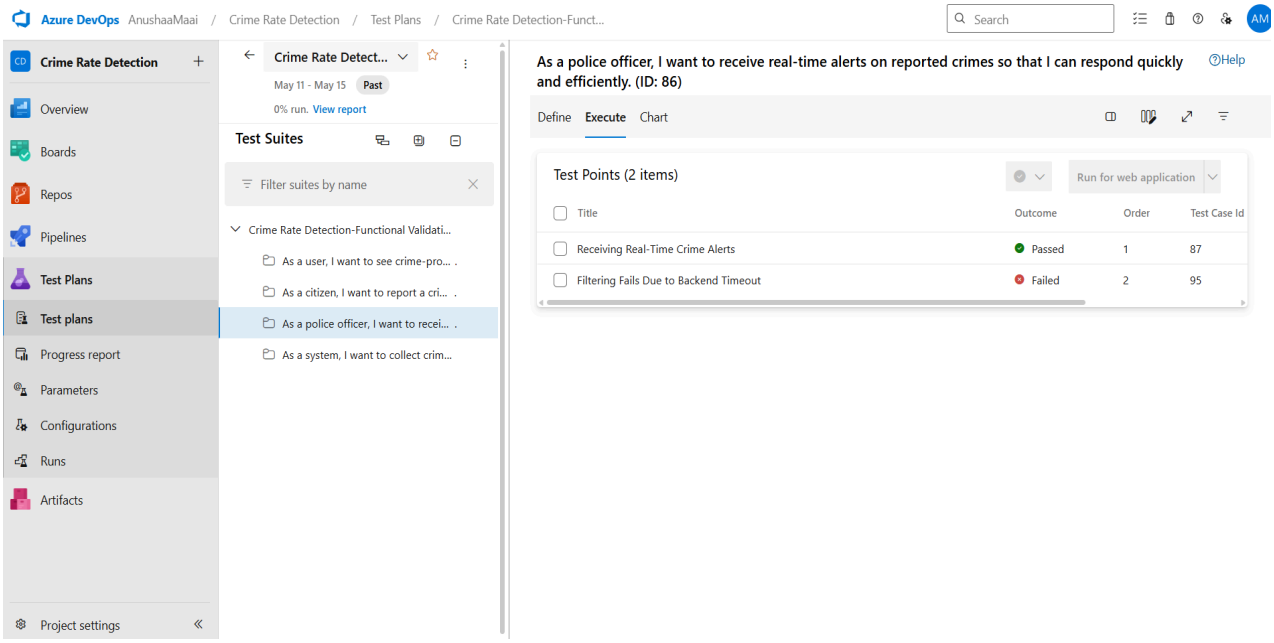
Test and feedback

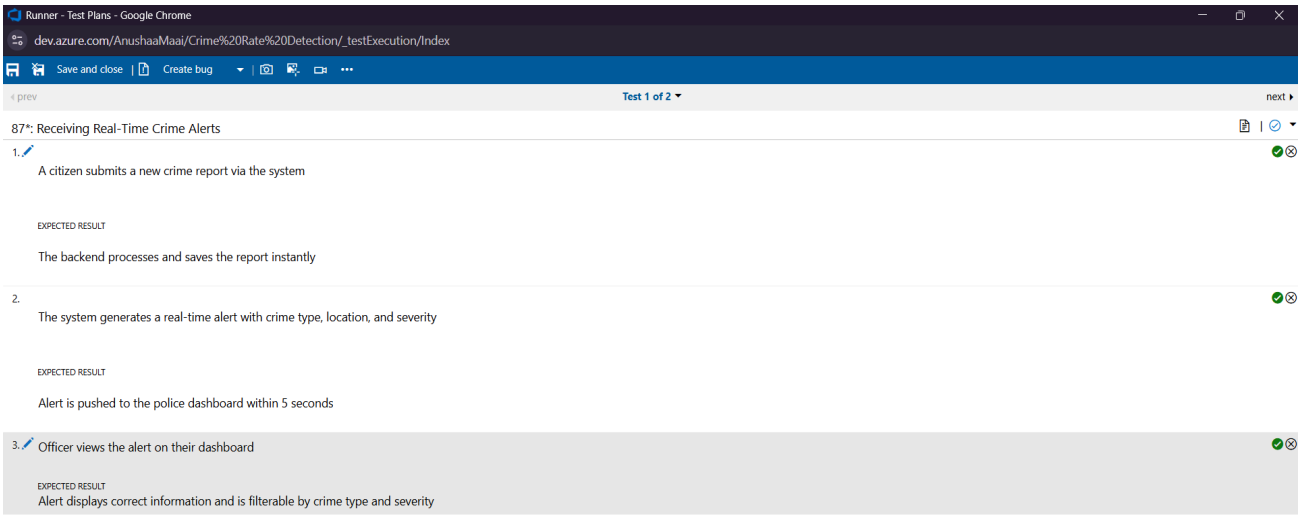


Showing it as an extension

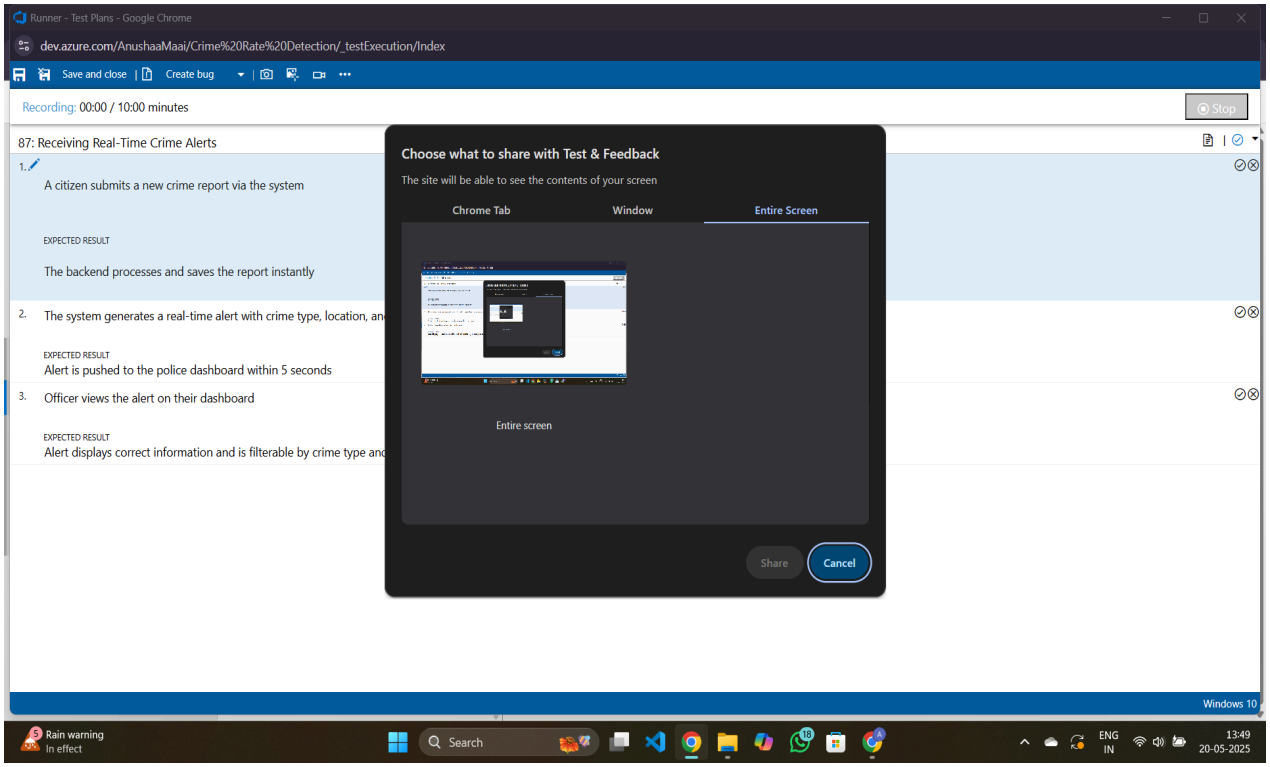


5. Running the test cases

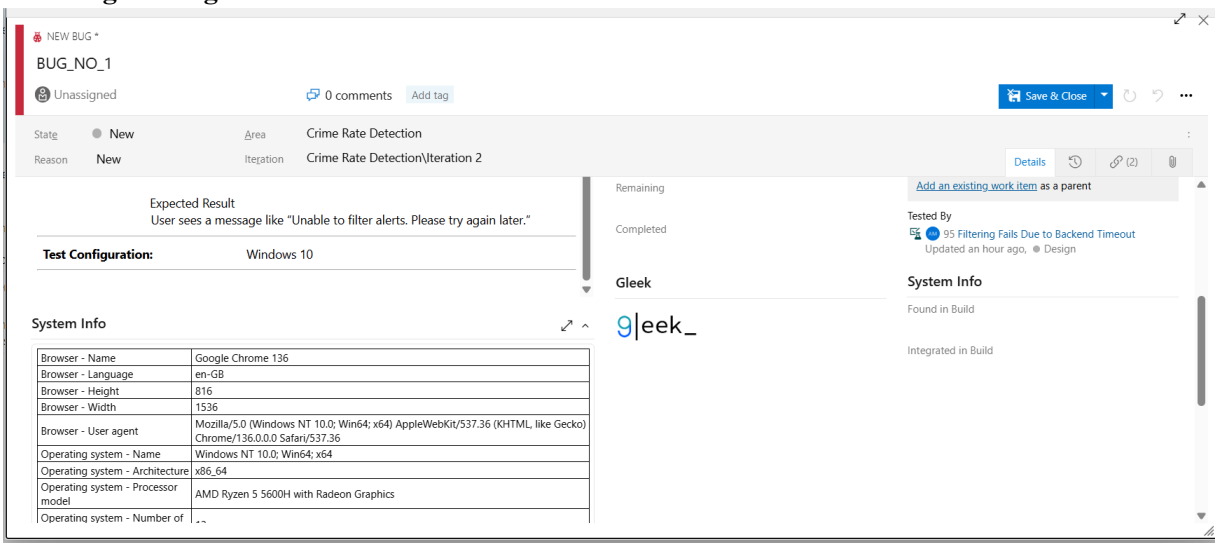


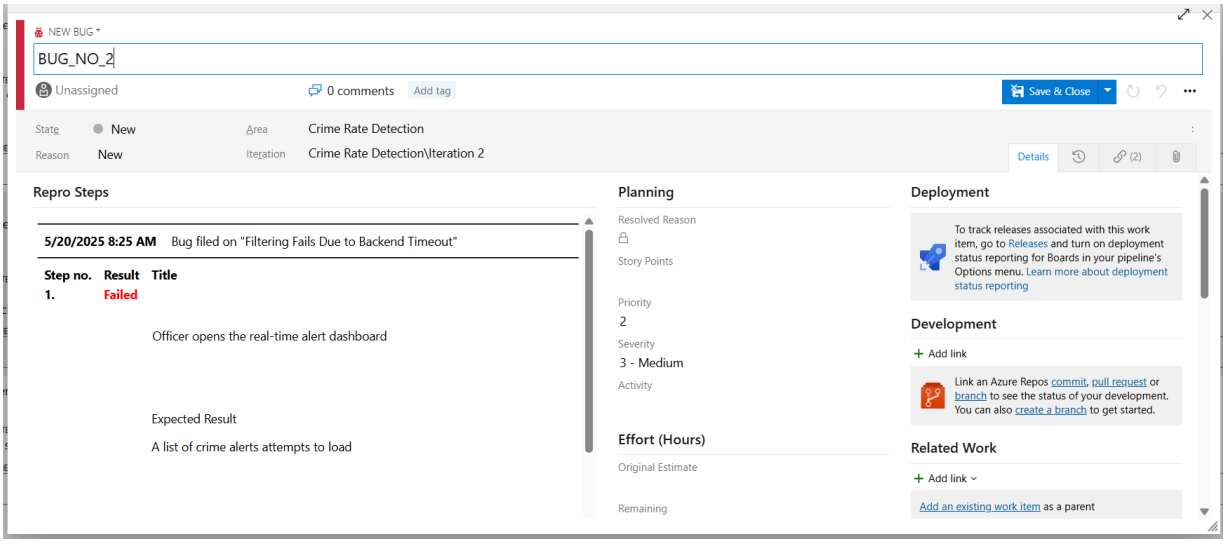


6. Recording the test case

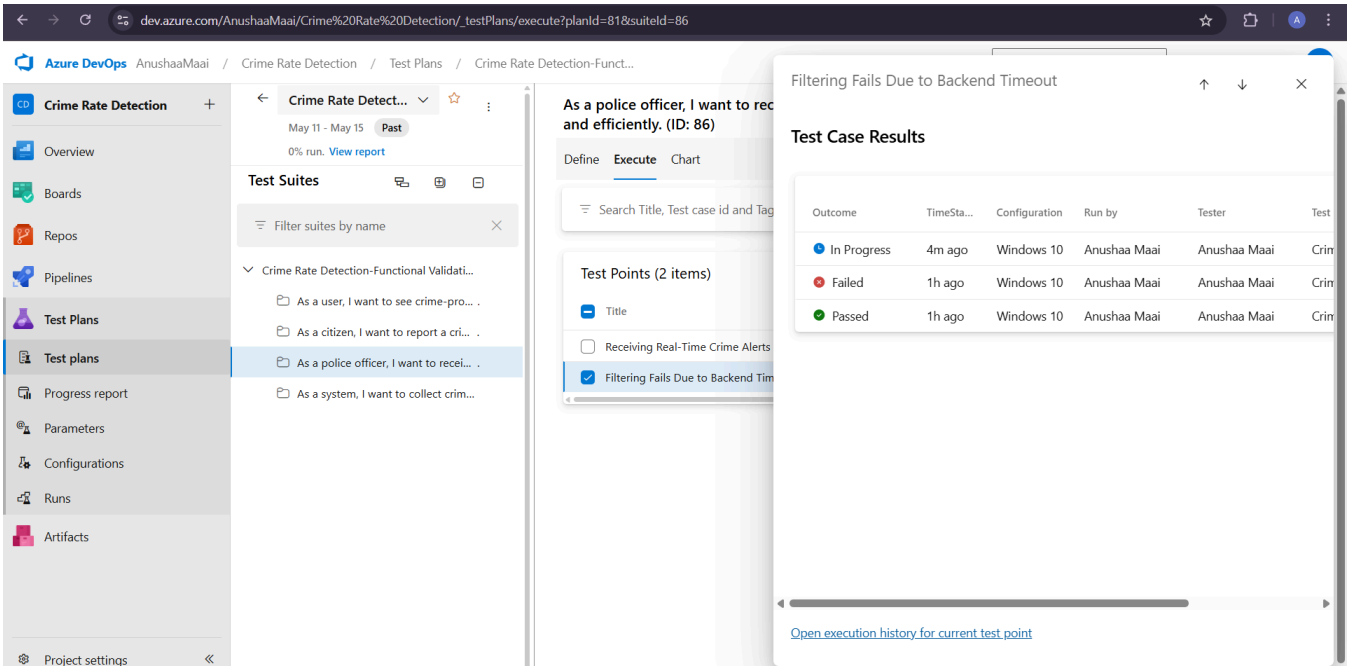


7. Creating the bug

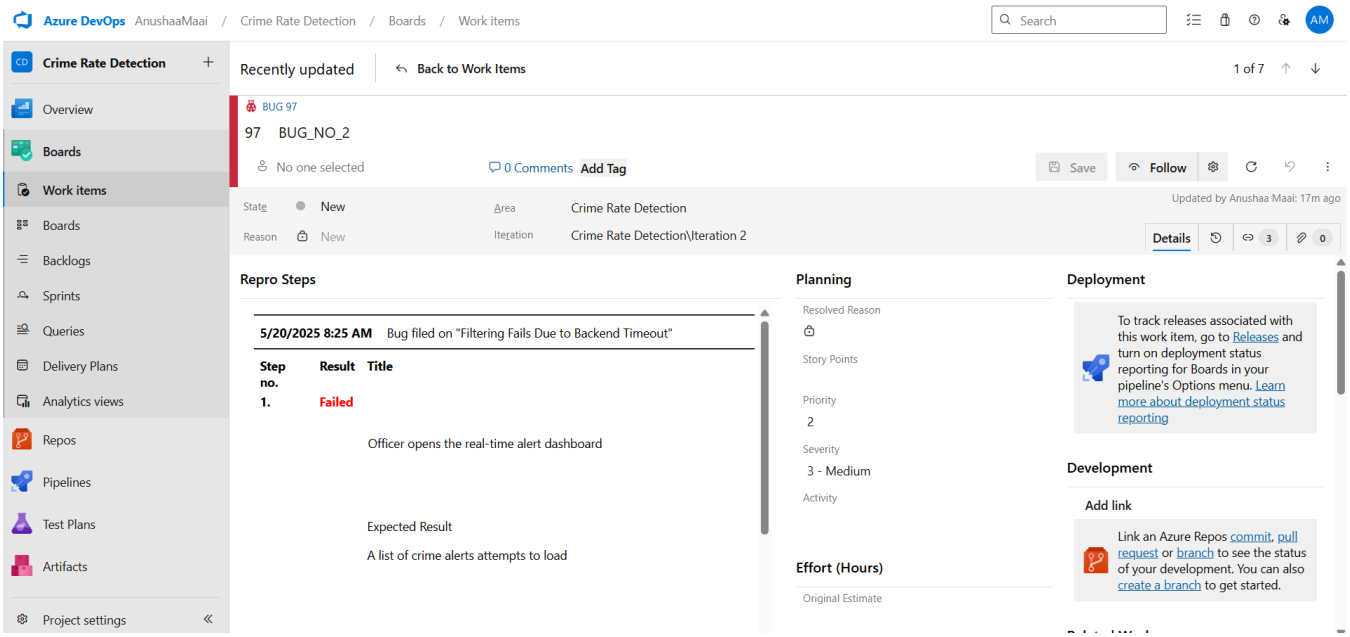




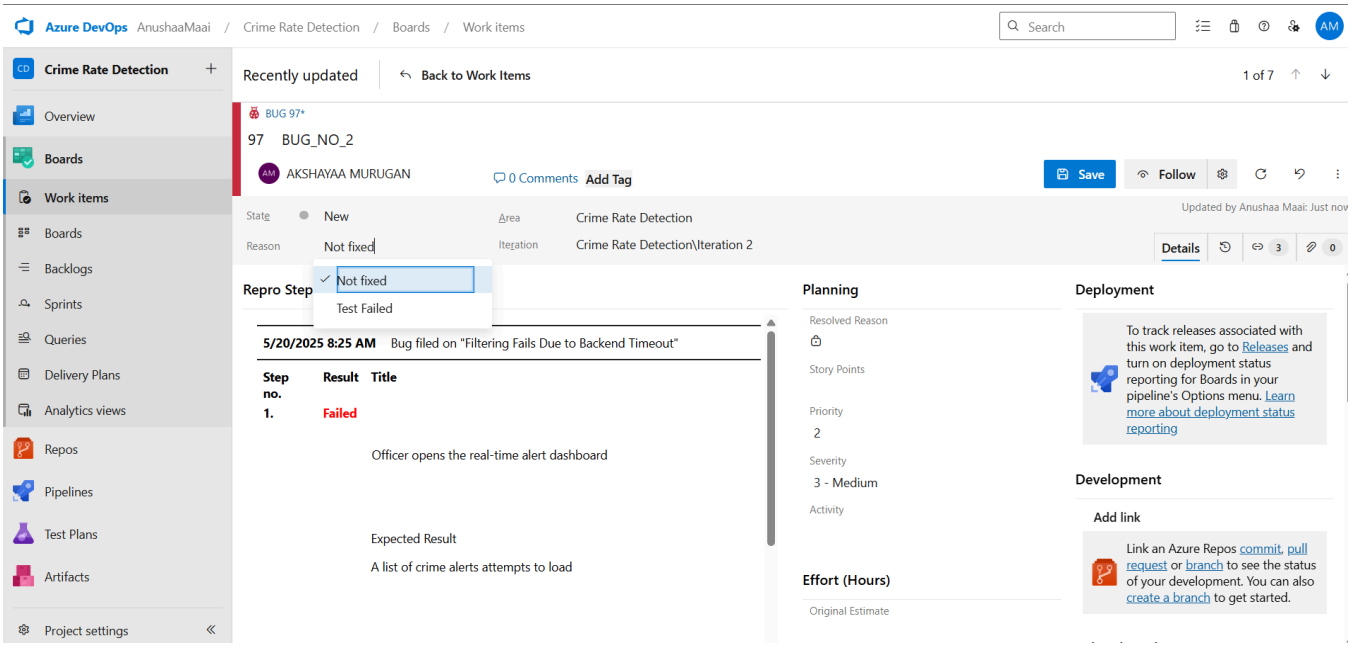
8. Test case results



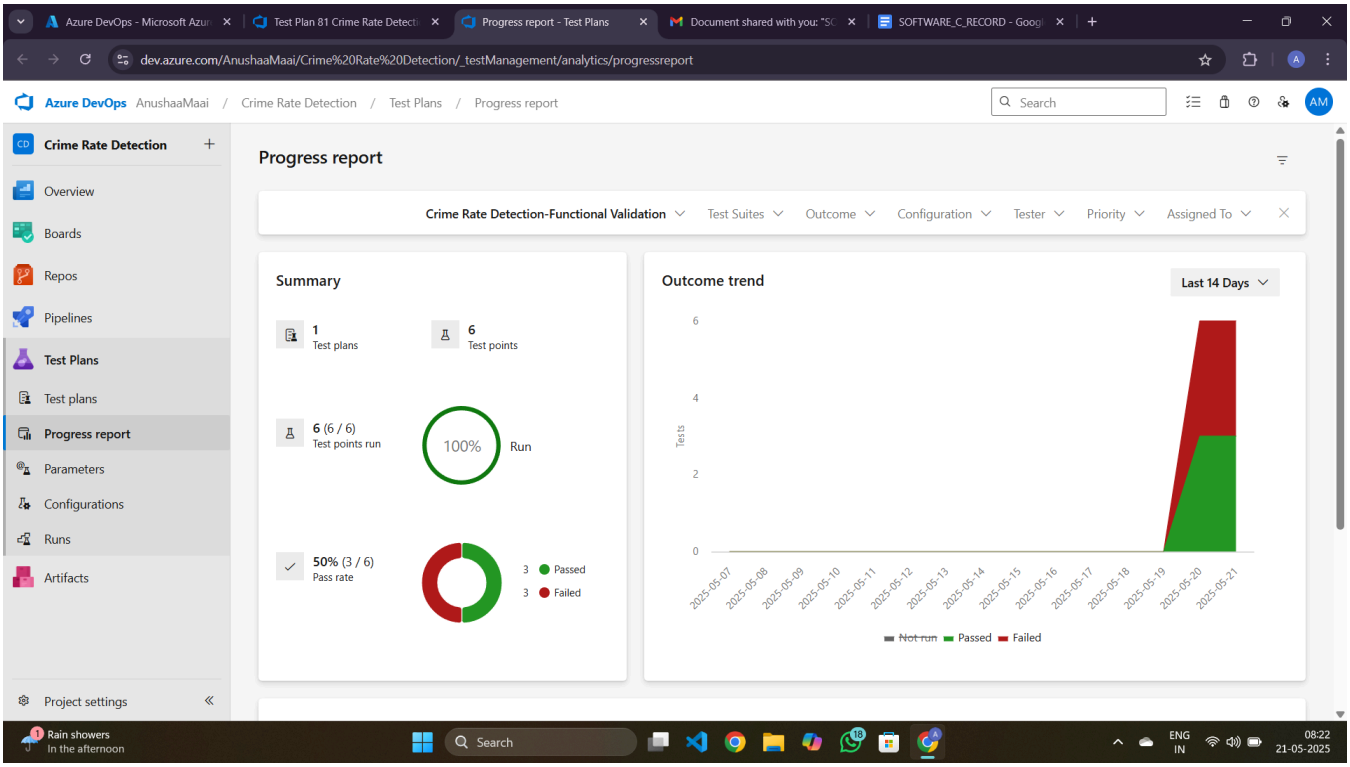
9. Test report summary

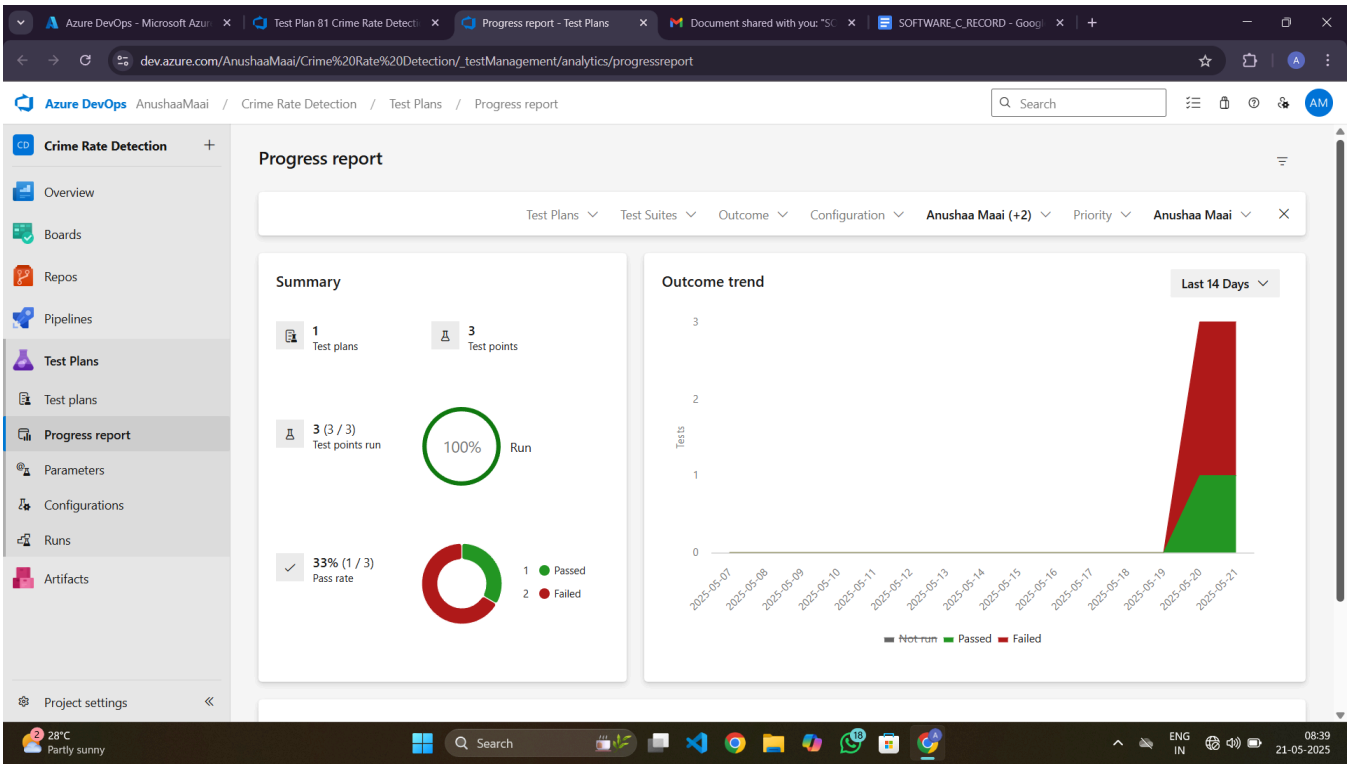


- Assigning bug to the developer and changing state

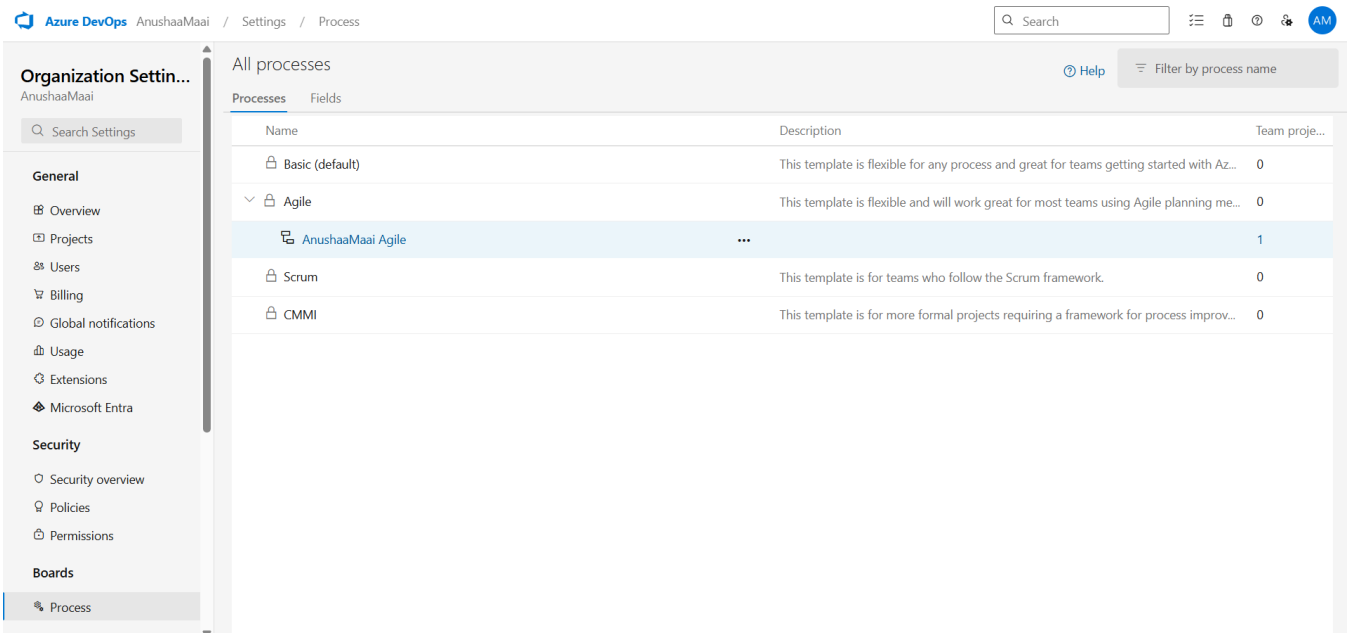


10. Progress report

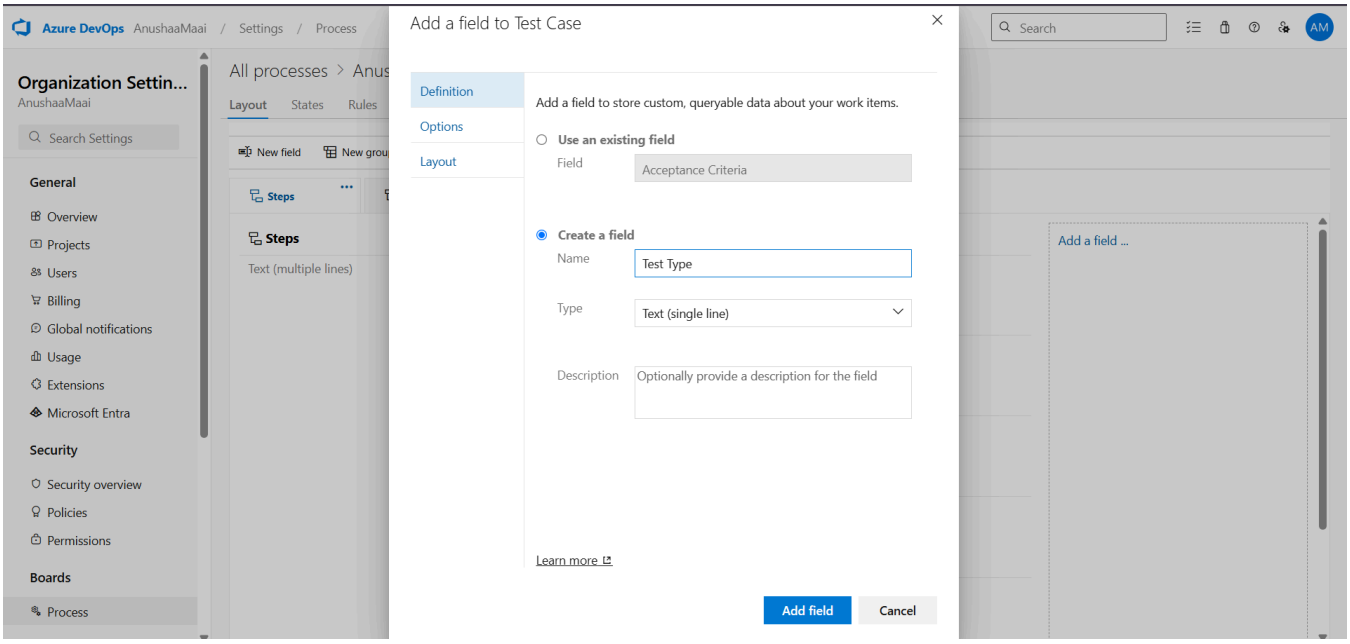


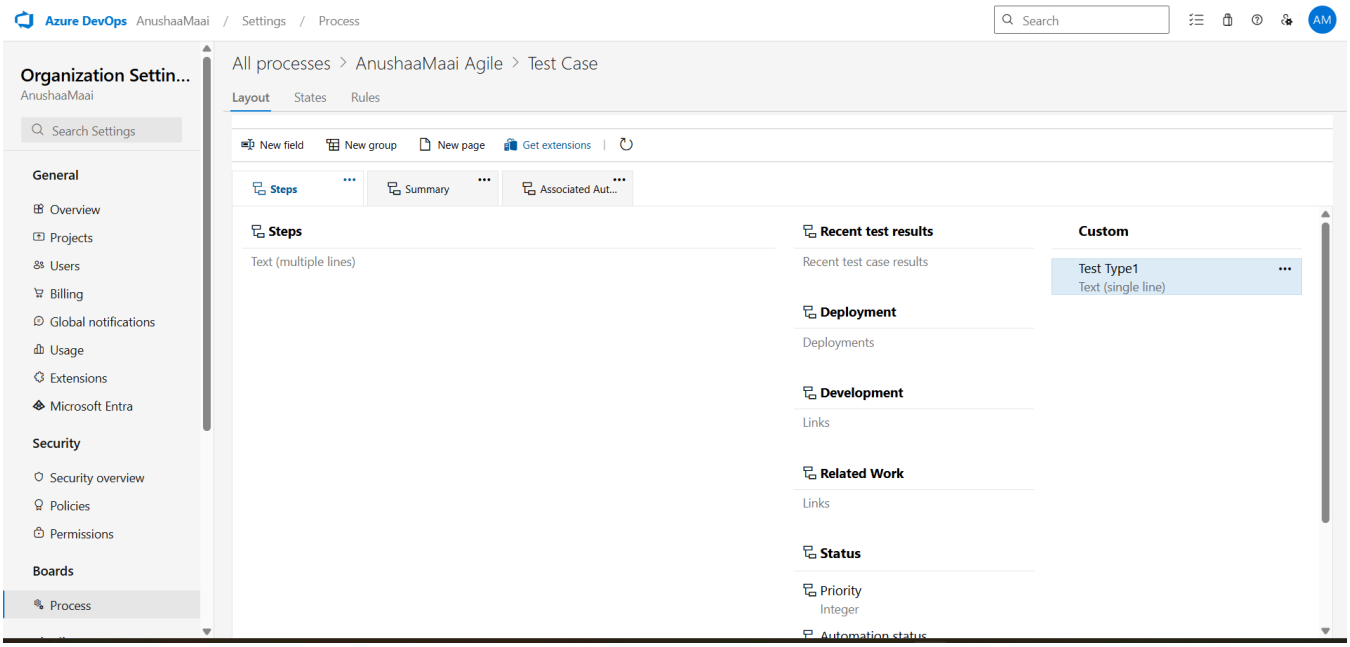


11. Changing the test template



12. View the new test case template





RESULT: The test plans and test cases for the user stories is created in Azure DevOps with Happy Path and Error Path

Exp.No: 9	LOAD TESTING AND PERFORMANCE TESTING
Date: 25/04/2025	

AIM: To create an Azure Load Testing resource and run a load test to evaluate the performance of a target endpoint.

LOAD TESTING

Steps to Create an Azure Load Testing Resource:

Before you run your first test, you need to create the Azure Load Testing resource:

1. Sign in to Azure Portal

   Go to https://portal.azure.com and log in.
2. Create the Resource o Go to *Create a resource* → Search for "Azure Load Testing".
   - o Select Azure Load Testing and click Create.
3. Fill in the Configuration Details

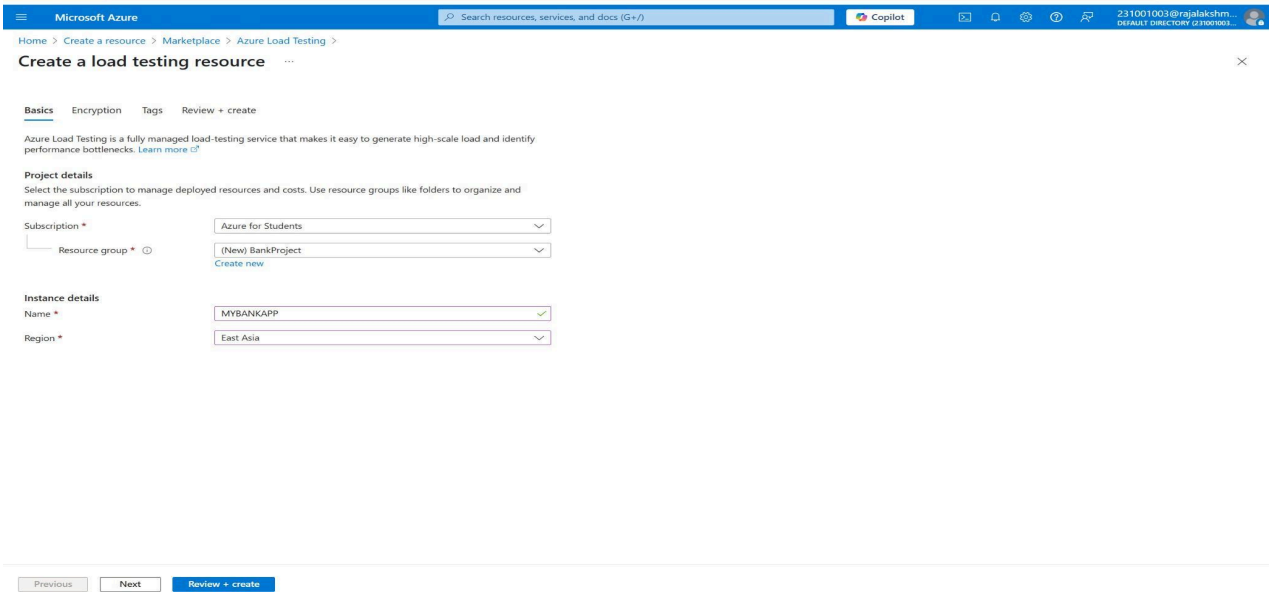
   o Subscriptio*n:* Choose your Azure subscription

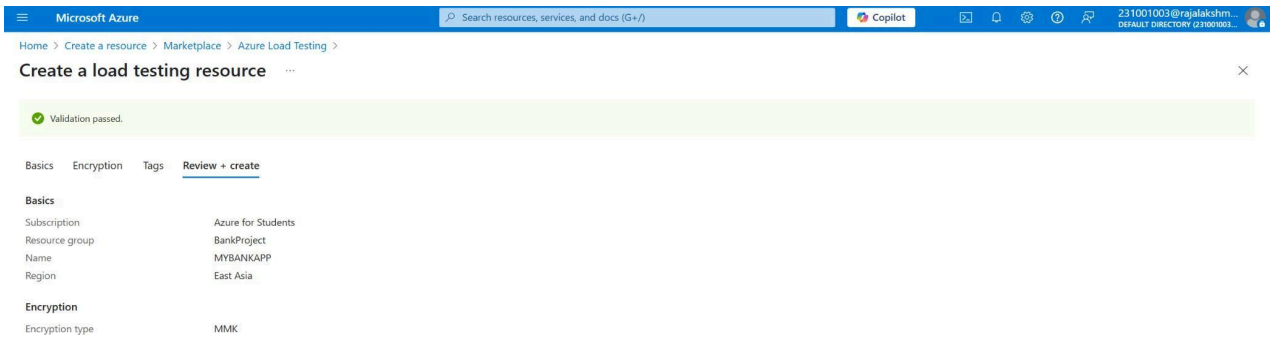
   o *Resource Group:* Create new or select an existing one.

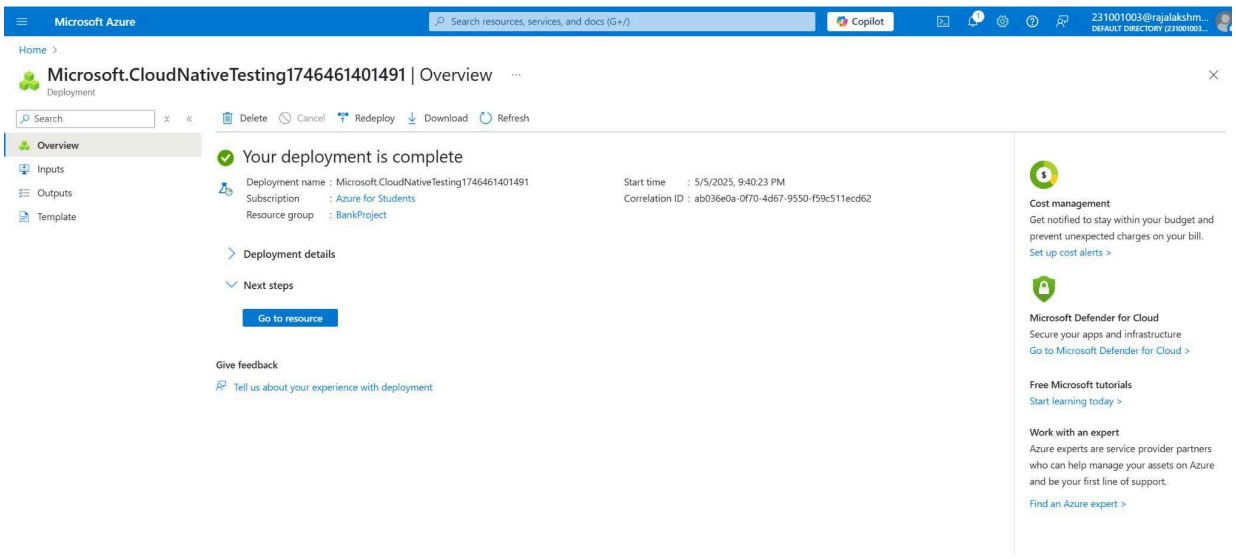
   o *Name:* Provide a unique name (no special characters).

   o *Location:* Choose the region for hosting the resource.

4. (Optional) Configure tags for categorization and billing.
5. Click Review + Create, then Create.
6. Once deployment is complete, click Go to resource.


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
231001003@rajalakshm...
DEFAULT DIRECTORY (231001003...
Home > Create a resource > Marketplace > Azure Load Testing >
Create a load testing resource
Basics Encryption Tags Review + create
Azure Load Testing is a fully managed load-testing service that makes it easy to generate high-scale load and identify performance bottlenecks. Learn more
Project details
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.
Subscription *
Azure for Students
Resource group *
(New) BankProject
Create new
Instance details
Name *
MYBANKAPP
Region *
East Asia
Previous
Next
Review + create

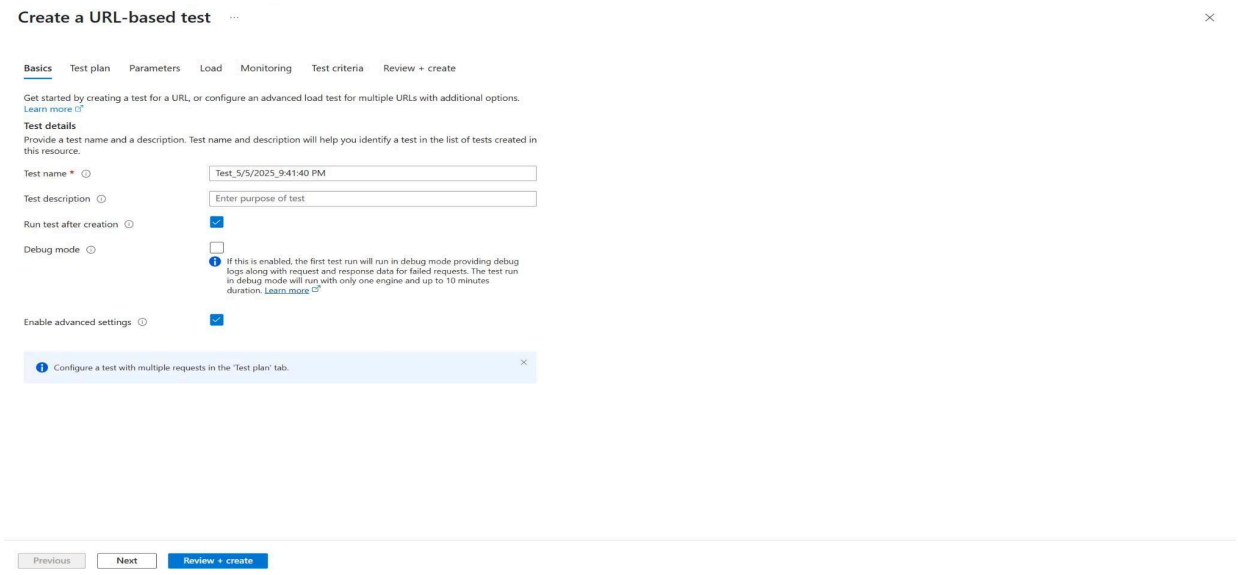

Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
231001003@rajalakshm...
DEFAULT DIRECTORY (231001003...
Home > Create a resource > Marketplace > Azure Load Testing >
Create a load testing resource
Validation passed.
Basics Encryption Tags Review + create
Basics
Subscription Azure for Students
Resource group BankProject
Name MYBANKAPP
Region East Asia
Encryption
Encryption type MMK

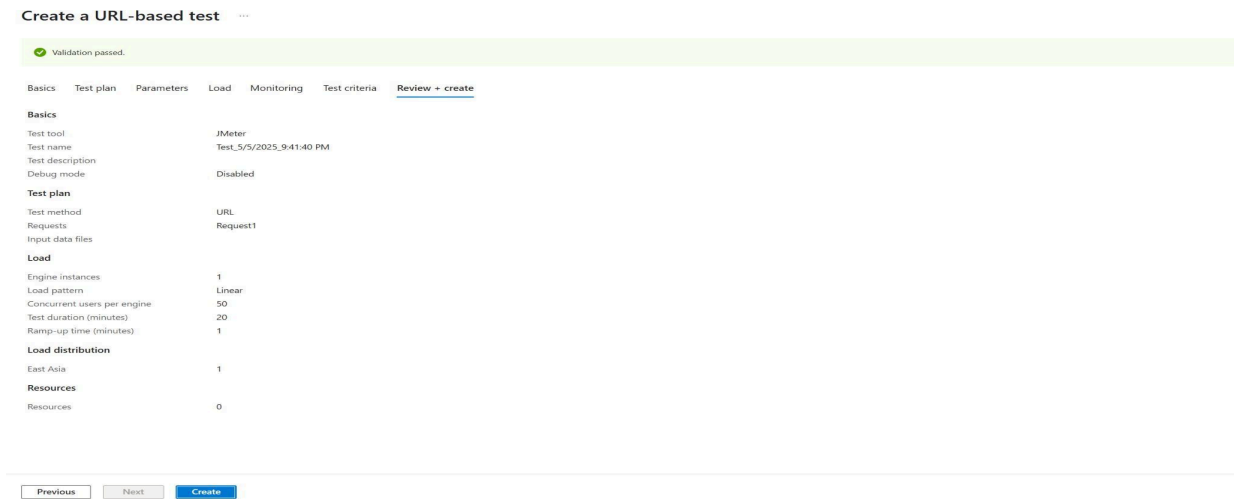

Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
231001003@rajalakshm...
DEFAULT DIRECTORY (231001003...
Home >
Microsoft.CloudNativeTesting1746461401491 | Overview
Deployment
Search
Overview
Inputs
Outputs
Template
Delete Cancel Redeploy Download Refresh
Your deployment is complete
Deployment name : Microsoft.CloudNativeTesting1746461401491
Subscription : Azure for Students
Resource group : BankProject
Start time : 5/5/2025, 9:40:23 PM
Correlation ID : ab036e0a-0f70-4d67-9550-f59c511ecd62
Deployment details
Next steps
Go to resource
Give feedback
Tell us about your experience with deployment
Cost management
Get notified to stay within your budget and prevent unexpected charges on your bill.
Set up cost alerts >
Microsoft Defender for Cloud
Secure your apps and infrastructure
Go to Microsoft Defender for Cloud >
Free Microsoft tutorials
Start learning today >
Work with an expert
Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.
Find an Azure expert >

Steps to Create and Run a Load Test:

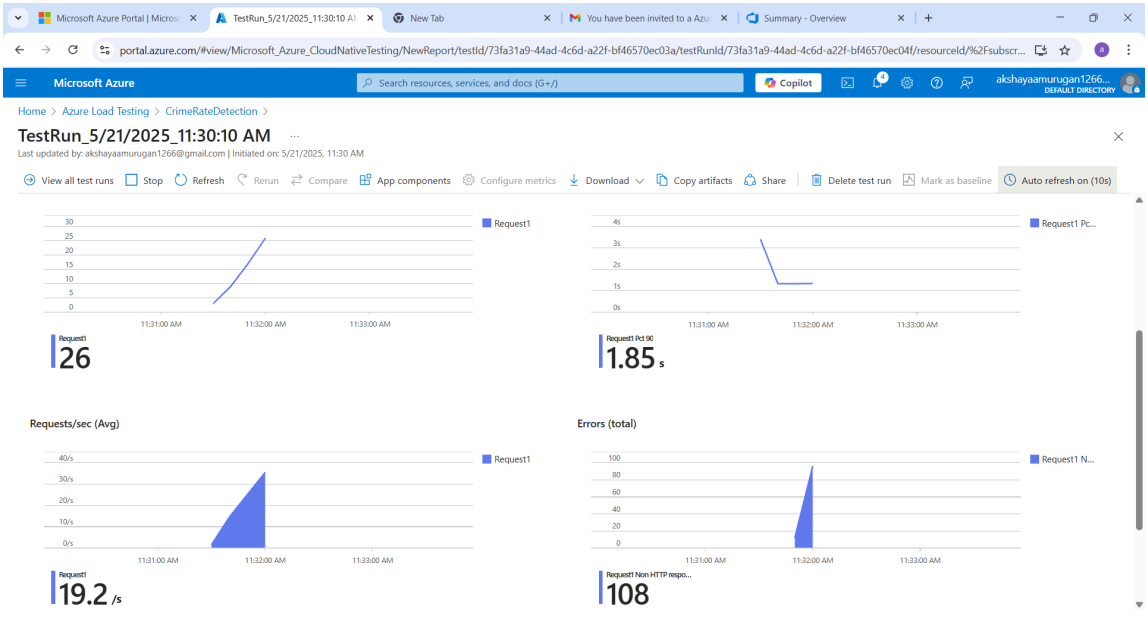
Once your resource is ready:

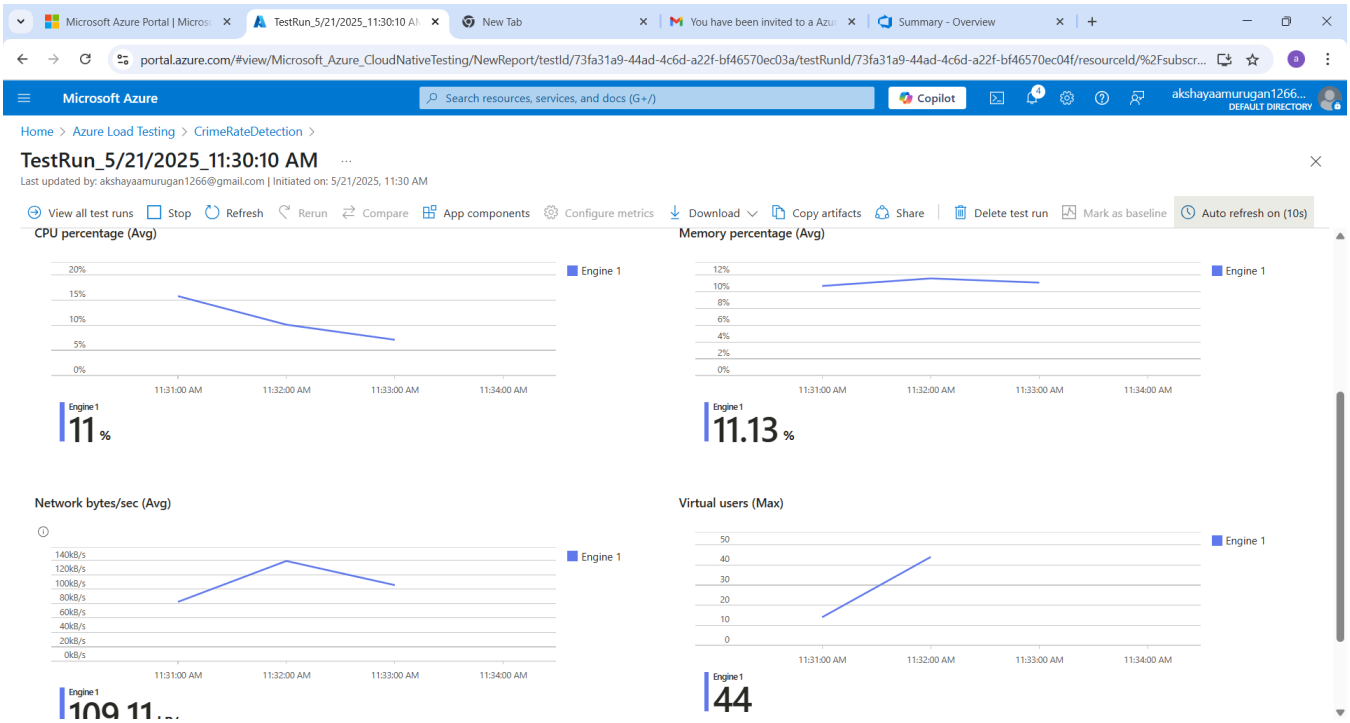
1. Go to your Azure Load Testing resource and click Add HTTP requests > Create.
2. Basics Tab o *Test Name:* Provide a unique name.
    - o *Description:* (Optional) Add test purpose.
    - o *Run After Creation:* Keep checked.
3. Load Settings o *Test URL:* Enter the target endpoint (e.g., https://yourapi.com/products).
4. Click Review + Create → Create to start the test.

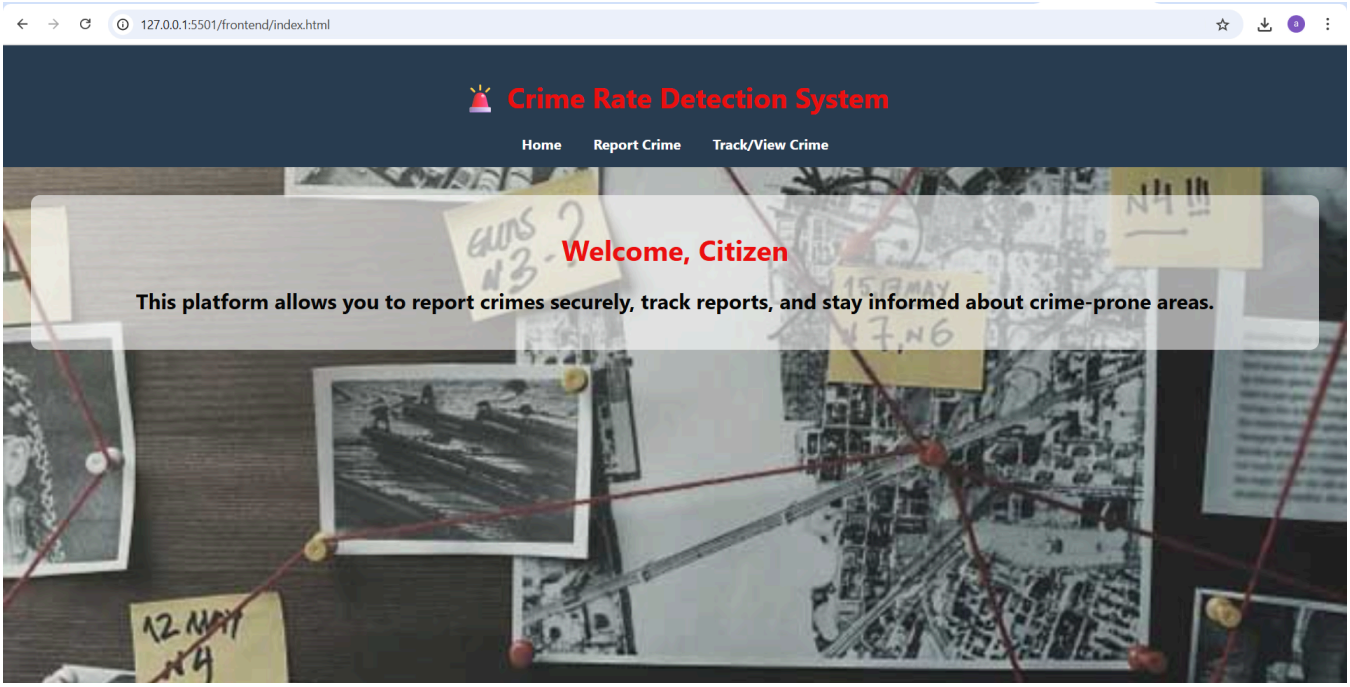


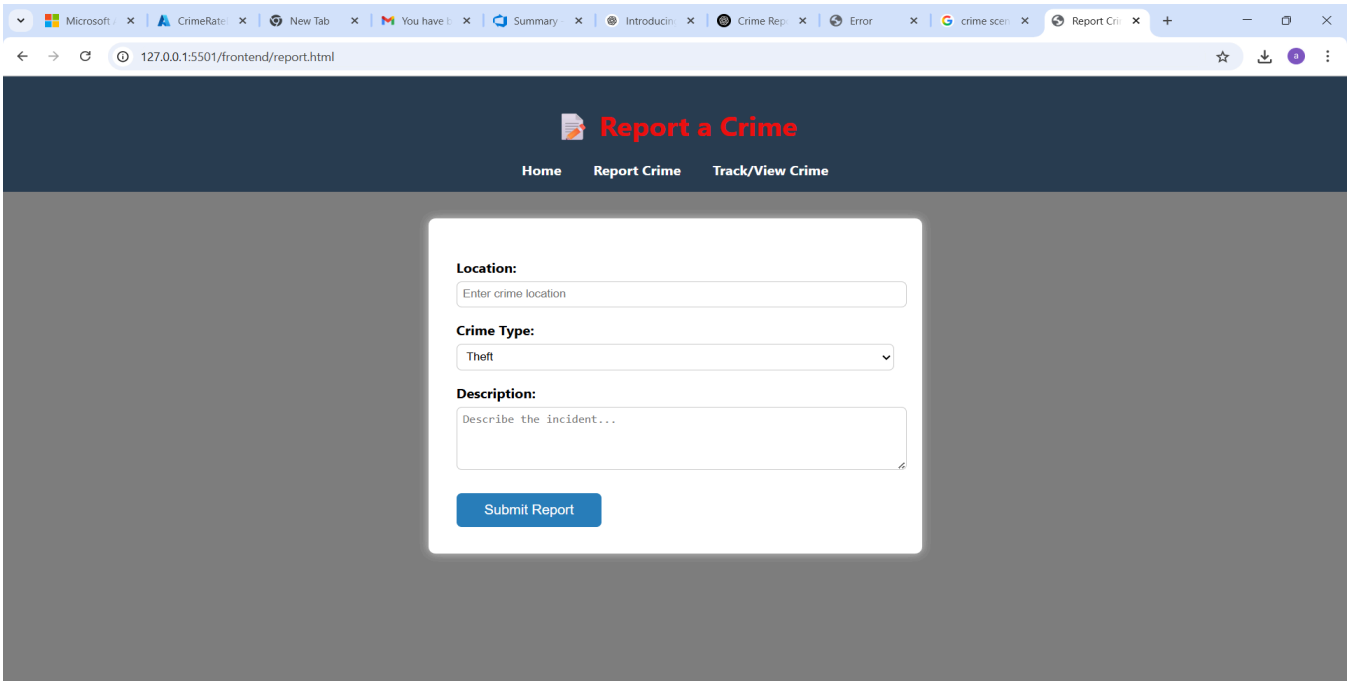


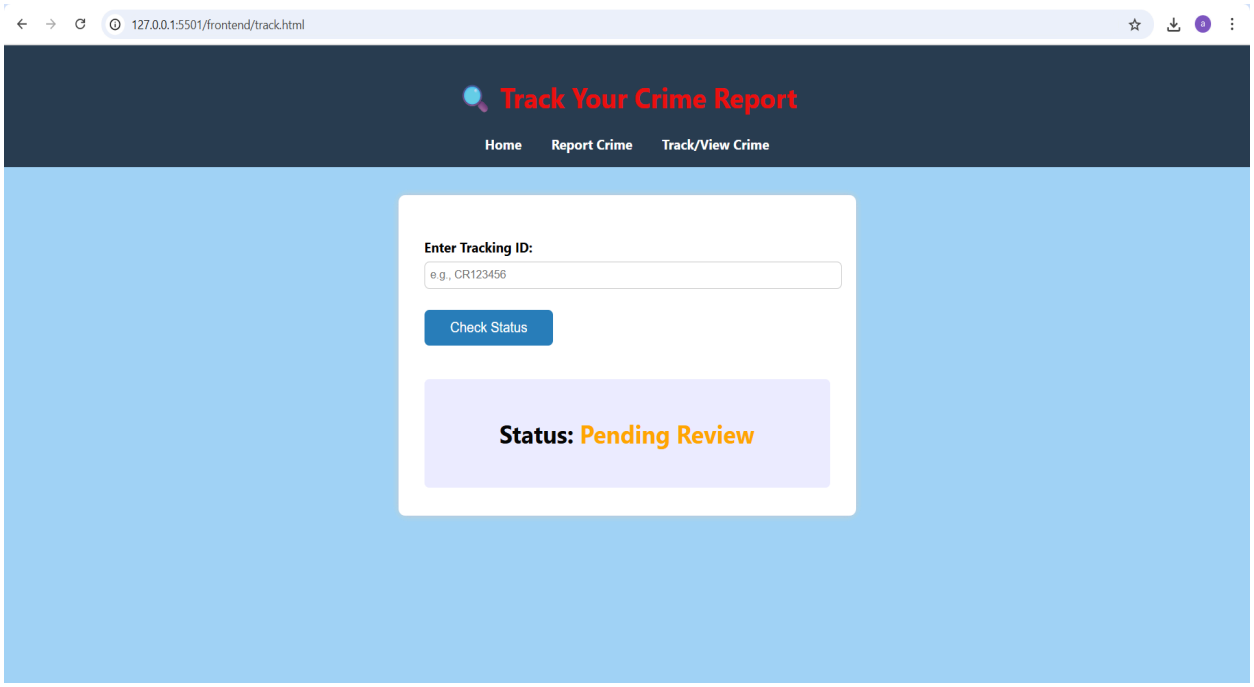
Load Testing






127.0.0.1:5501/frontend/index.html
Crime Rate Detection System
Home
Report Crime
Track/View Crime
Welcome, Citizen
This platform allows you to report crimes securely, track reports, and stay informed about crime-prone areas.


127.0.0.1:5501/frontend/report.html
Report a Crime
Home
Report Crime
Track/View Crime
Location:
Enter crime location
Crime Type:
Theft
Description:
Describe the incident...
Submit Report

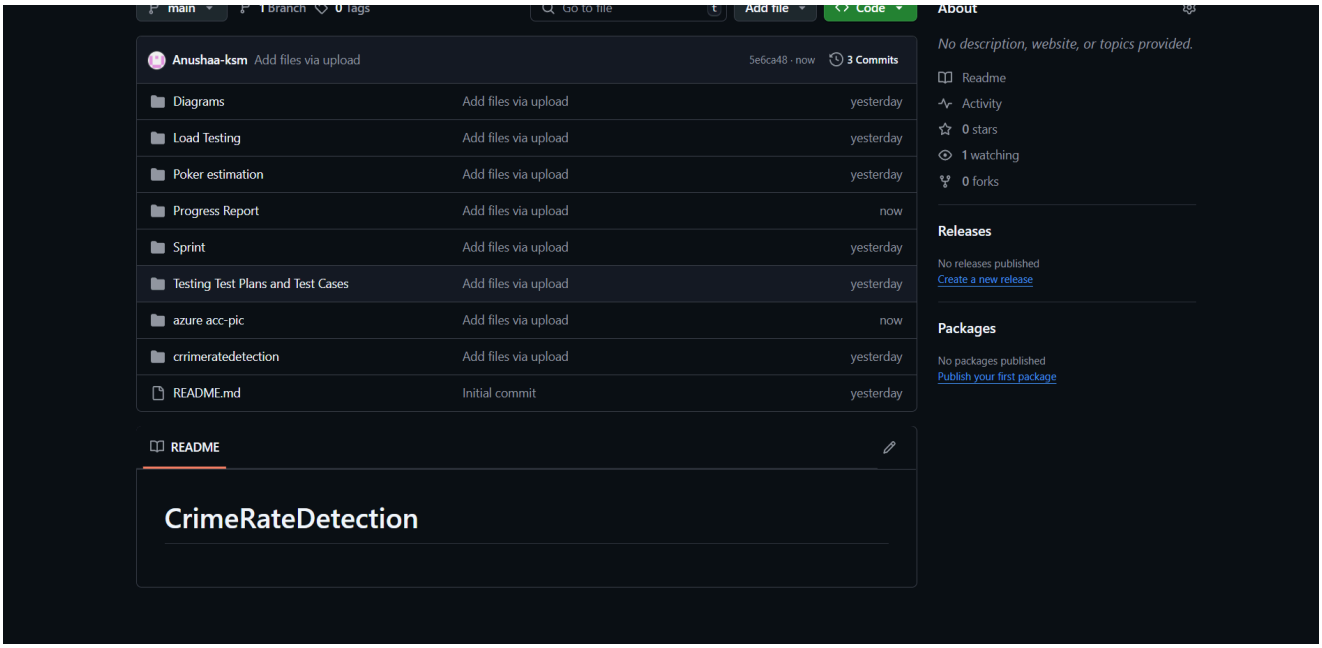


RESULT: Successfully created the Azure Load Testing resource and executed a load test to assess the performance of the specified endpoint.

Exp.No: 10	GITHUB PROJECT STRUCTURE & NAMING CONVENTIONS
Date: 02/05/2025	

AIM: To provide a clear and organized view of the project's folder structure and file naming conventions, helping contributors and users easily understand, navigate, and extend the Crime Rate Detection project.

GITHUB PROJECT STRUCTURE



RESULT: The GitHub repository clearly displays the organized project structure and consistent naming conventions, making it easy for users and contributors to understand and navigate the codebase.