

**EXP NO: 1**

## **AZURE DEVOPS ENVIRONMENT SETUP**

### **Aim:**

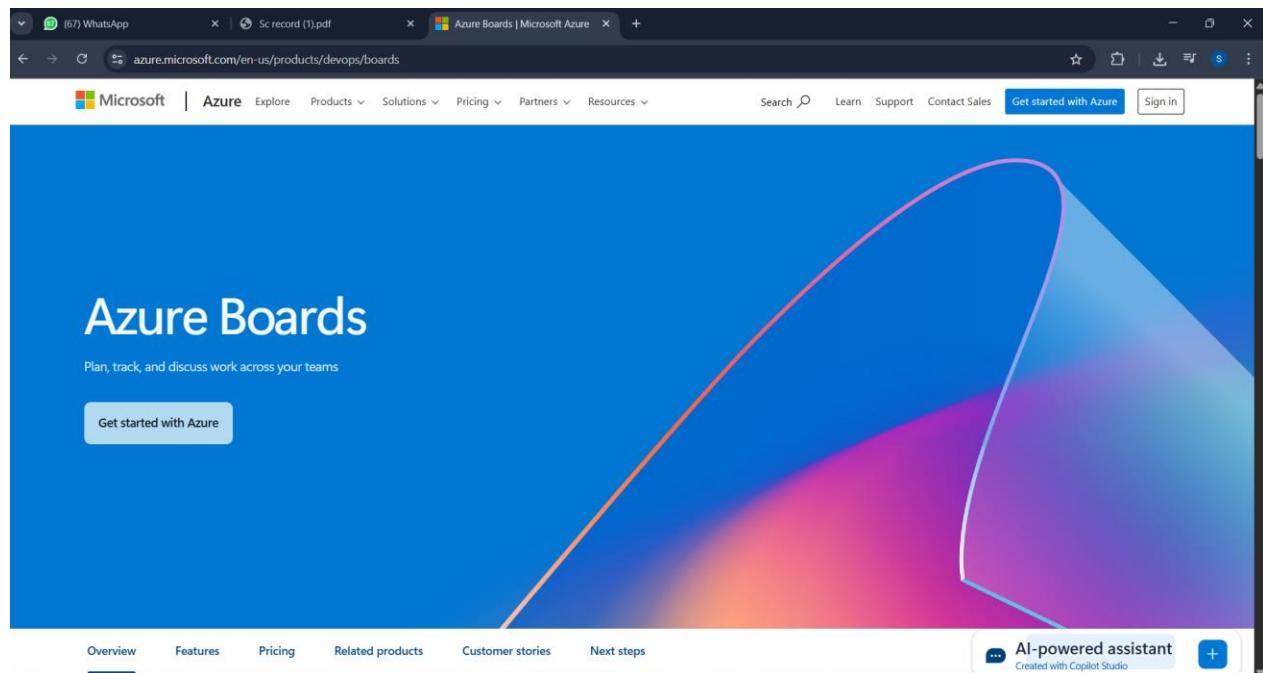
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

### **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

The screenshot shows the Microsoft Azure home page. At the top, there's a search bar with the placeholder "Search resources, services, and docs (G+/-)". Below the search bar, the "Welcome to Azure!" message is displayed, along with three promotional cards: "Start with an Azure free trial", "Manage Microsoft Entra ID", and "Azure for Students". Under "Azure services", there are icons for "Create a resource", "Azure DevOps organizations", "Azure Database for MySQL...", "Devices", "Quickstart Center", "Azure AI services", "Kubernetes services", "Virtual machines", "App Services", and "More services". A "Resources" section is also visible.

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

The screenshot shows the Microsoft Azure home page with a search bar containing the text "devops". The search results are displayed under the "Services" tab, listing items like "Azure Native New Relic Service", "Managed DevOps Pools", and "Azure DevOps organizations". Other tabs include "All", "Services (7)", and "Resources". The right side of the page remains consistent with the standard Azure home page layout, featuring the "Welcome to Azure!", "Azure services" section, and the "Resources" section.

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.

← → ⌂ portal.azure.com/#view/AzureTfsExtension/OrganizationsTemplateBlade

Microsoft Azure Search resources, services, and docs (G+)

Home > Azure DevOps ...

We've made it easier to manage Azure DevOps billing and subscriptions. You can [set up billing](#), change your subscription or pay for more users and resources within Azure DevOps. [Learn more](#)

## Azure DevOps

Plan smarter, collaborate better, and ship faster with a set of modern dev services

[My Azure DevOps Organizations](#)

[Get started using Azure DevOps](#)  
Billing management for Azure DevOps

[Give feedback](#)

[Tell us about your experience with the Azure DevOps page](#)



### Result:

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

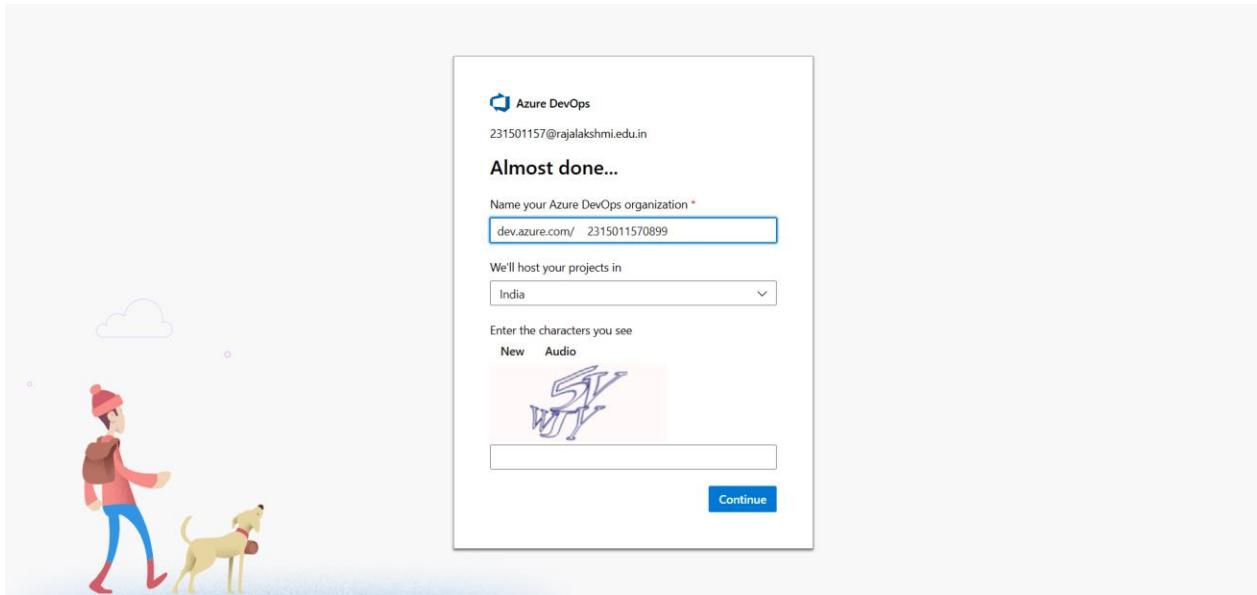
**EXP NO: 2**

## **AZURE DEVOPS PROJECT SETUP AND USER STORY MANAGEMENT**

### **Aim:**

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

#### **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.

The screenshot shows the 'Create a project to get started' page. At the top right are search and navigation icons. Below is a large heading 'Create a project to get started'. A 'Project name \*' input field is followed by a 'Description' input field. Under 'Visibility', 'Public' is disabled and 'Private' is selected. A note states that public projects are disabled for the organization. Below this is an 'Advanced' section with dropdowns for 'Version control' (Git) and 'Work item process' (Agile). A 'Create project' button is at the bottom.

Search

≡ ⚡ 🌐 ⓘ ⚙️ SS

## Create a project to get started

Project name \*

Description

Visibility

Public

Anyone on the internet can view the project. Certain features like TFVC are not supported.

Private

Only people you give access to will be able to view this project.

Advanced

Version control

Work item process

Git

Agile

+ Create 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.

The screenshot shows the Azure DevOps Organizations dashboard. At the top right, there's a sign-out link for 'Sonia J S'. On the left, there's a profile picture placeholder with 'SS' and some contact information: 'Sonia J S', '231501157@rajalakshmi.edu.in', 'India', and '231501157@rajalakshmi.edu.in'. Below this is a section for 'Visual Studio Dev Essentials' with a brief description and a 'Use your benefits' button. The main area is titled 'Azure DevOps Organizations' and lists several organizations: 'dev.azure.com/231501157' (Owner), 'dev.azure.com/2315011570466' (Owner), and 'dev.azure.com/2315011570509' (Owner). Under 'dev.azure.com/2315011570509', there's a 'Projects' section with 'Crime Rate Detector' and an 'Actions' section with a 'Open in Visual Studio' button. There are also links for 'New project' and other organization members.

#### 4. Project dashboard

The screenshot shows the Azure DevOps project dashboard for 'Crime Rate Detector'. The left sidebar has a navigation menu with 'Overview' selected. The main content area is titled 'Crime Rate Detector' and includes sections for 'About this project' (with a placeholder for adding a project description), 'Project stats' (showing 12 work items and 0 work items), and 'Members' (listing three users: SS, Sonja, and Priya). The top navigation bar shows the URL 'dev.azure.com/2315011570509/Crime%20Rate%20Detector' and includes a search bar and various navigation icons.

5. To manage user stories:

- 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.
- 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 the new User Story.

The screenshot shows the Azure DevOps Boards page for the 'Crime Rate Detector' project. The left sidebar is the navigation menu. The main area displays the backlog for the 'Crime Rate Detector Team'. The backlog table has columns for Order, Work Item Type, Title, State, Effort, Business Area, and Tags. Three epics are listed:

Order	Work Item Type	Title	State	Effort	Business Area	Tags
1	Epic	> Data Collection and Preparation	Active		Business	
2	Epic	> Model Building and Evaluation	Active		Business	
3	Epic	> Crime Prediction and Forecasting	Active		Business	

The screenshot shows the Microsoft sign-in page. The user profile is displayed on the right, showing the name 'Sonia J S', the email '231501157@rajalakshmi.edu.in', and links to 'My Microsoft account' and 'Switch directory'. A large red circular icon with the letters 'SS' is overlaid on the profile picture. At the bottom, there is a 'Sign in with a different account' link and a placeholder for a business card.

**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**

## **Aim:**

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

## **Create Epic, Features, User Stories, Task**

The screenshot shows the Azure DevOps Boards backlog view for the 'Crime Rate Detector' project. The backlog is organized into three main epics:

- Epic 1: Data Collection and Preparation
  - Feature: Crime Data Collection
  - User Story: As a data engineer, I want to collect crime data from ...
    - Task: Identify reliable data sources (e.g., police records, p...)
    - Task: Automate data scraping or API integration.
  - User Story: As a data analyst, I want to extract time, location, and ...
    - Task: Parse and structure the extracted data
    - Task: Store data in a clean and consistent format (e.g., CS...)
  - Feature: Data Cleaning and Preprocessing
- Epic 2: Model Building and Evaluation
  - Feature: Algorithm Selection and Training
  - Feature: Ensemble and Optimization
- Epic 3: Crime Prediction and Forecasting

## **1. Fill in Epics**

The screenshot shows the Azure DevOps Work items creation page for creating a new epic. The 'Title' field is highlighted with an error message: "NEW EPIC \* Field 'Title' cannot be empty." The work item form includes fields for State (New), Reason (New), Area (Crime Rate Detector), Iteration (Crime Rate Detector\Sprint 1), Description (Click to add Description.), Planning (Priority 2, Risk), Deployment (Release notes), and Development (Add link).

## 2.Fill in Features

The screenshot shows the Azure DevOps interface for creating a new work item. The URL is [https://dev.azure.com/2315011570509/Crime%20Rate%20Detector/\\_workitems/create/Feature](https://dev.azure.com/2315011570509/Crime%20Rate%20Detector/_workitems/create/Feature). The left sidebar is titled "Crime Rate Detector" and includes options like Overview, Boards, Work items, and Project settings. The main area is titled "Work Items" and shows a "NEW FEATURE" message: "Field 'Title' cannot be empty." A red error icon is next to the title field. The form fields include:

- Description:** No one selected, 0 Comments, Add Tag.
- Planning:** State: New, Reason: New, Area: Crime Rate Detector, Iteration: Crime Rate Detector\Sprint 1, Priority: 2, Risk: 2, Business Value: 1, Time Criticality: 1, Start Date: Select a date..., Target Date: Select a date... .
- 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.
- Discussion:** Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request. A "switch to Markdown editor" link is present.

## 3.Fill in User Story Details

The screenshot shows the Azure DevOps interface for creating a new work item. The URL is [https://dev.azure.com/2315011570509/Crime%20Rate%20Detector/\\_workitems/create/User%20Story](https://dev.azure.com/2315011570509/Crime%20Rate%20Detector/_workitems/create/User%20Story). The left sidebar is titled "Crime Rate Detector" and includes options like Overview, Boards, Work items, and Project settings. The main area is titled "Work Items" and shows a "NEW USER STORY" message: "Field 'Title' cannot be empty." A red error icon is next to the title field. The form fields include:

- Description:** Click to add Description.
- Acceptance Criteria:** Click to add Acceptance Criteria.
- Planning:** State: New, Reason: New, Area: Crime Rate Detector, Iteration: Crime Rate Detector\Sprint 1, Story Points: 2, Priority: 2, Risk: 2.
- Classification:** Value area: Business.
- 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.
- Discussion:** Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request. A "switch to Markdown editor" link is present.

**Result:**

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

**EXP NO: 4**

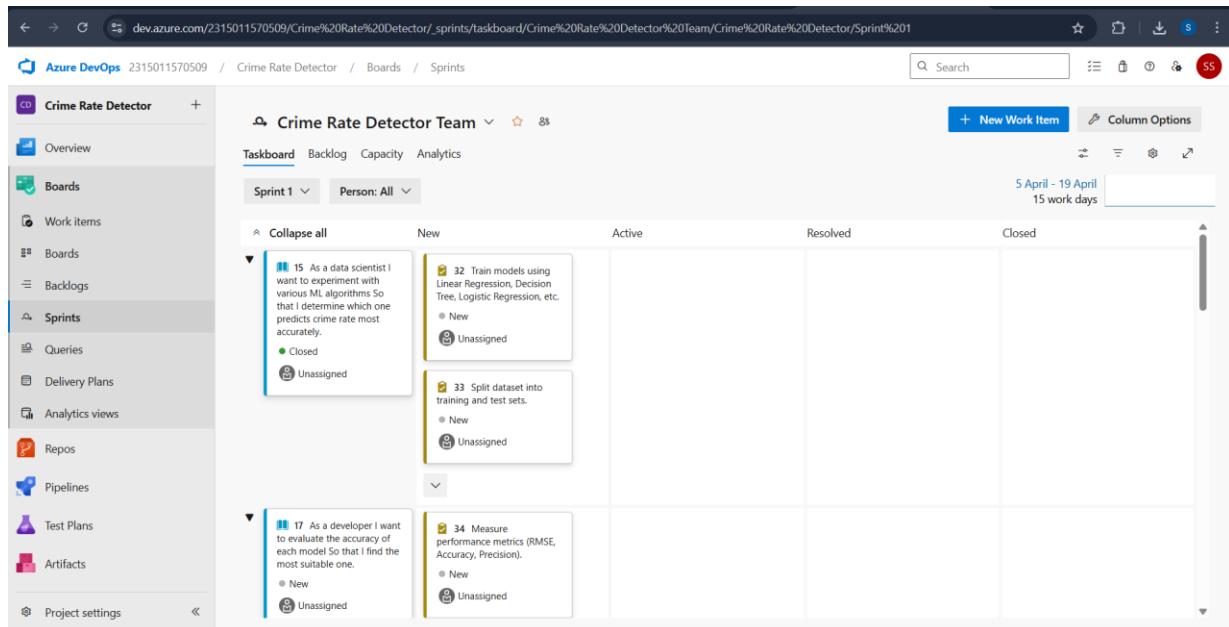
# SPRINT PLANNING

## Aim:

To assign user story to specific sprint for the Crime Rate Detector App Project.

## Sprint Planning

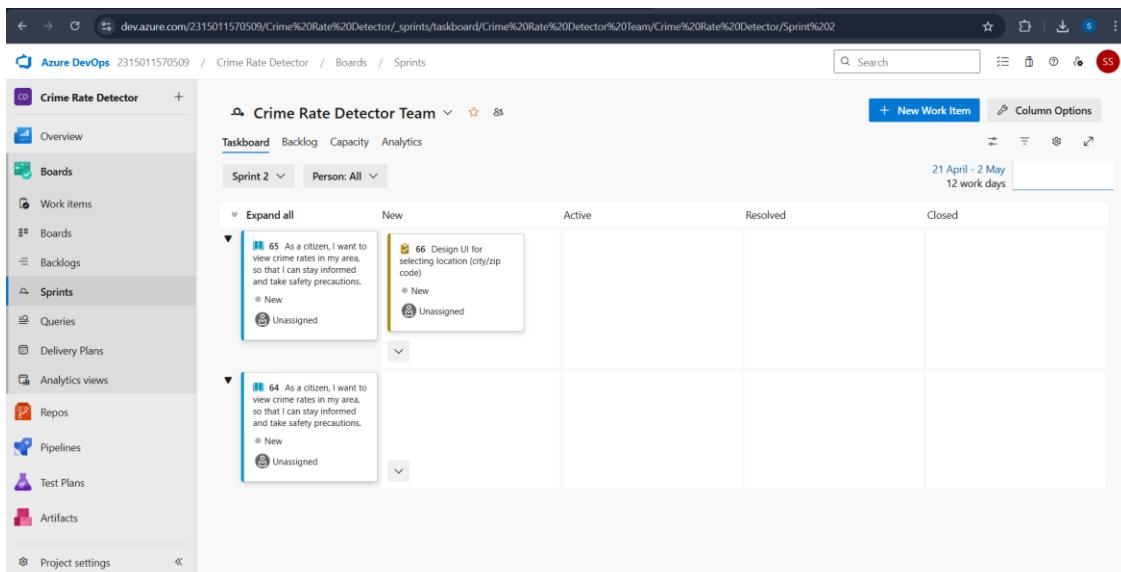
### Sprint 1



The screenshot shows the Azure DevOps Taskboard for the 'Crime Rate Detector' project under the 'Crime Rate Detector Team'. The current sprint is 'Sprint 1' (5 April - 19 April, 15 work days). The backlog is visible on the left, and the taskboard grid shows several user stories assigned to different columns (New, Active, Resolved, Closed). One user story is highlighted in the 'New' column:

User Story ID	Description	Status
15	As a data scientist I want to experiment with various ML algorithms So that I determine which one predicts crime rate most accurately.	New
32	Train models using Linear Regression, Decision Tree, Logistic Regression, etc.	Unassigned
33	Split dataset into training and test sets.	Unassigned
17	As a developer I want to evaluate the accuracy of each model So that I find the most suitable one.	New
34	Measure performance metrics (RMSE, Accuracy, Precision).	Unassigned

### Sprint 2



The screenshot shows the Azure DevOps Taskboard for the 'Crime Rate Detector' project under the 'Crime Rate Detector Team'. The current sprint is 'Sprint 2' (21 April - 2 May, 12 work days). The backlog is visible on the left, and the taskboard grid shows several user stories assigned to different columns (New, Active, Resolved, Closed). One user story is highlighted in the 'New' column:

User Story ID	Description	Status
65	As a citizen, I want to view crime rates in my area, so that I can stay informed and take safety precautions.	New
66	Design UI for selecting location (city/zip code)	Unassigned
64	As a citizen, I want to view crime rates in my area, so that I can stay informed and take safety precautions.	New

## Sprint 3

The screenshot shows the Azure DevOps Taskboard for the 'Crime Rate Detector' project under the 'Crime Rate Detector Team'. The board is set to 'Sprint 3' and 'Person: All'. The columns are 'New', 'Active', 'Resolved', and 'Closed'. There are four work items visible:

- 70** As a system admin, I want to manage user roles and data access, so that only authorized personnel can make critical changes.
  - Status: New
  - Assignee: Unassigned
- 67** As a law enforcement officer, I want to analyze crime trends over time, so I can identify patterns and hotspots.
  - Status: New
  - Assignee: Unassigned
- 71** Implement user authentication (login/signup).
  - Status: Active
  - Assignee: Sonia J S
- 68** Create dashboard layout for trend visualization.
  - Status: Closed
  - Assignee: Unassigned

### Result:

The Sprints are created for the Crime Rate Detector App Project.

**EXP NO: 5**

# **POKER ESTIMATION**

## **Aim:**

Create Poker Estimation for the user stories - Crime Rate Detector App Project.

## **Poker Estimation**

The screenshot shows the Azure DevOps interface for a work item titled "USER STORY 11". The work item details are as follows:

- Description:** 11 As a data engineer, I want to collect crime data from various sources so that I can create a comprehensive dataset.
- Assignee:** Sonia J S
- Comments:** 0
- Add Tag:** Add Tag
- State:** Resolved
- Reason:** Code complete and uni...
- Area:** Crime Rate Detector
- Iteration:** Crime Rate Detector

**Planning:**

- Story Points: 2
- Priority: 2
- Risk: 1

**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.

**Classification:**

- Value area: Business

**Discussion:**

Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request.

## **Result:**

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

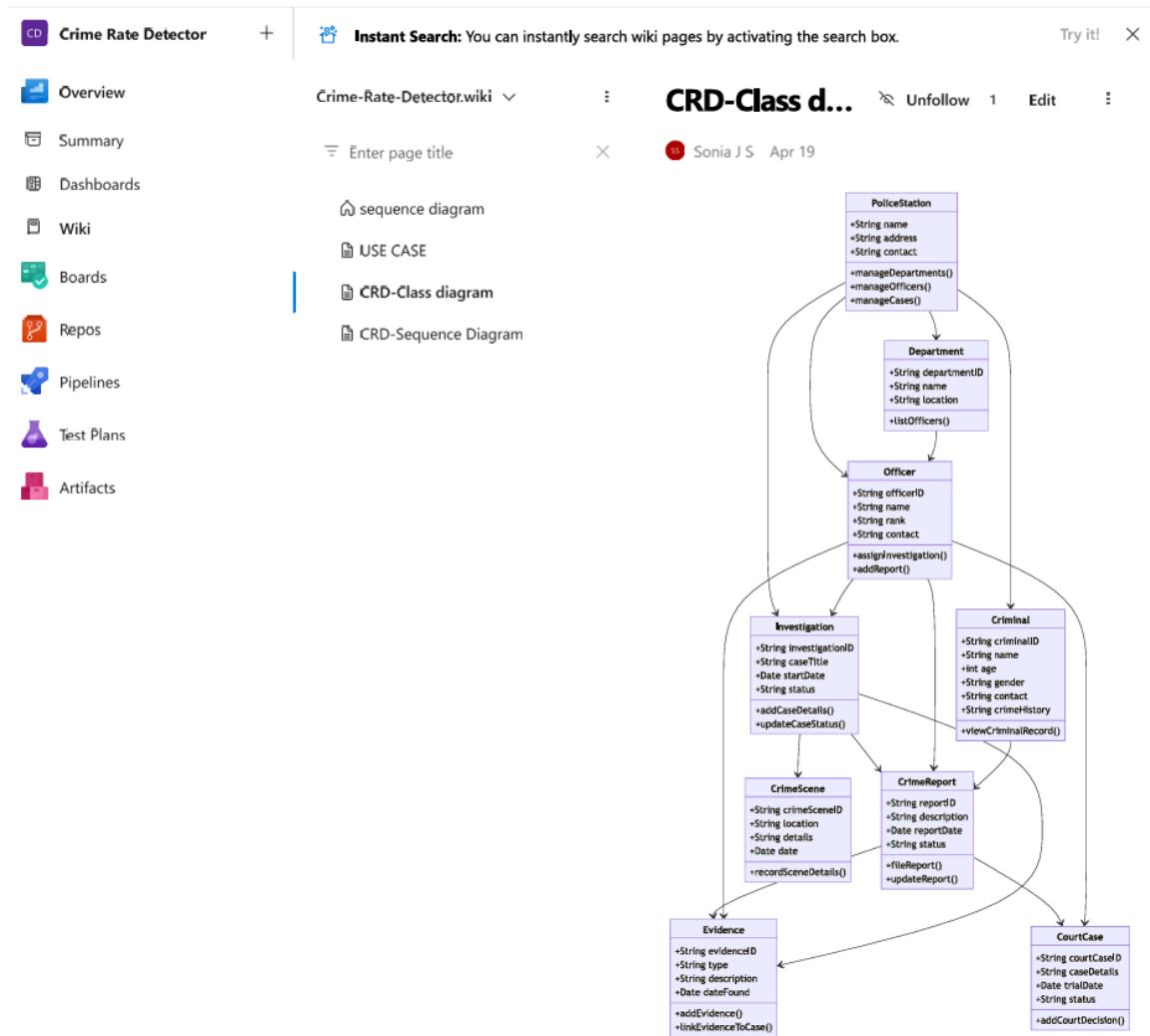
EXP NO: 6

# DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

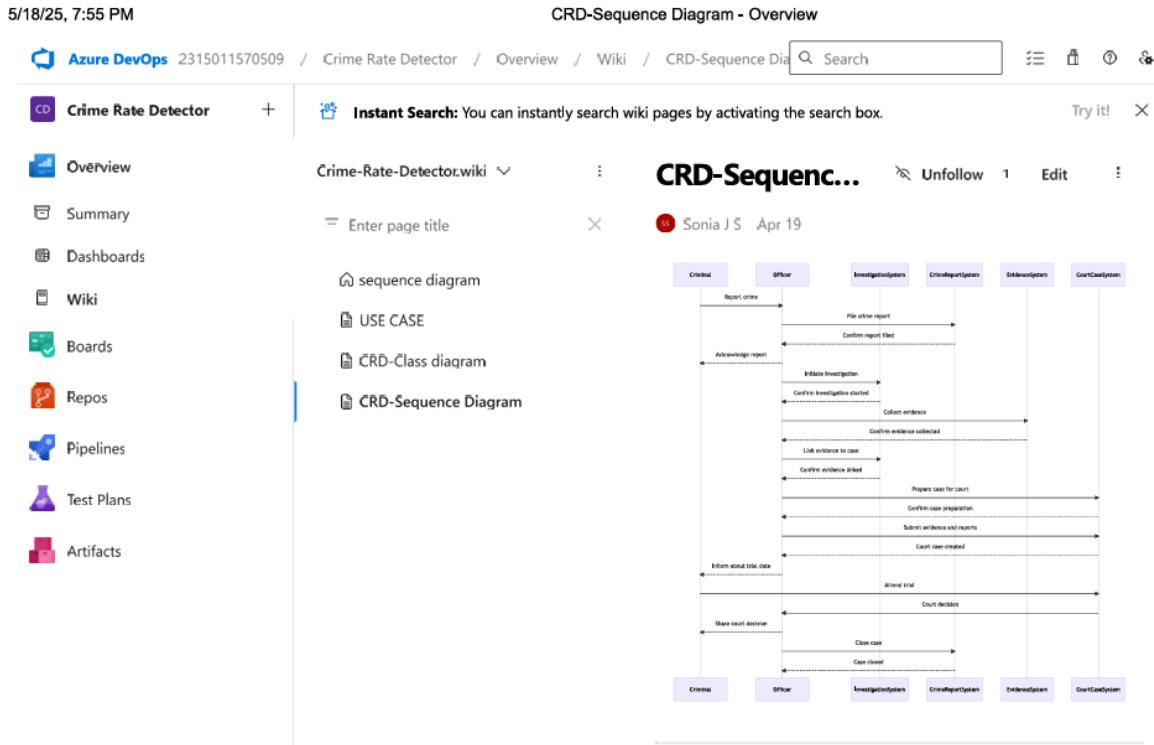
## Aim:

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 Detector App

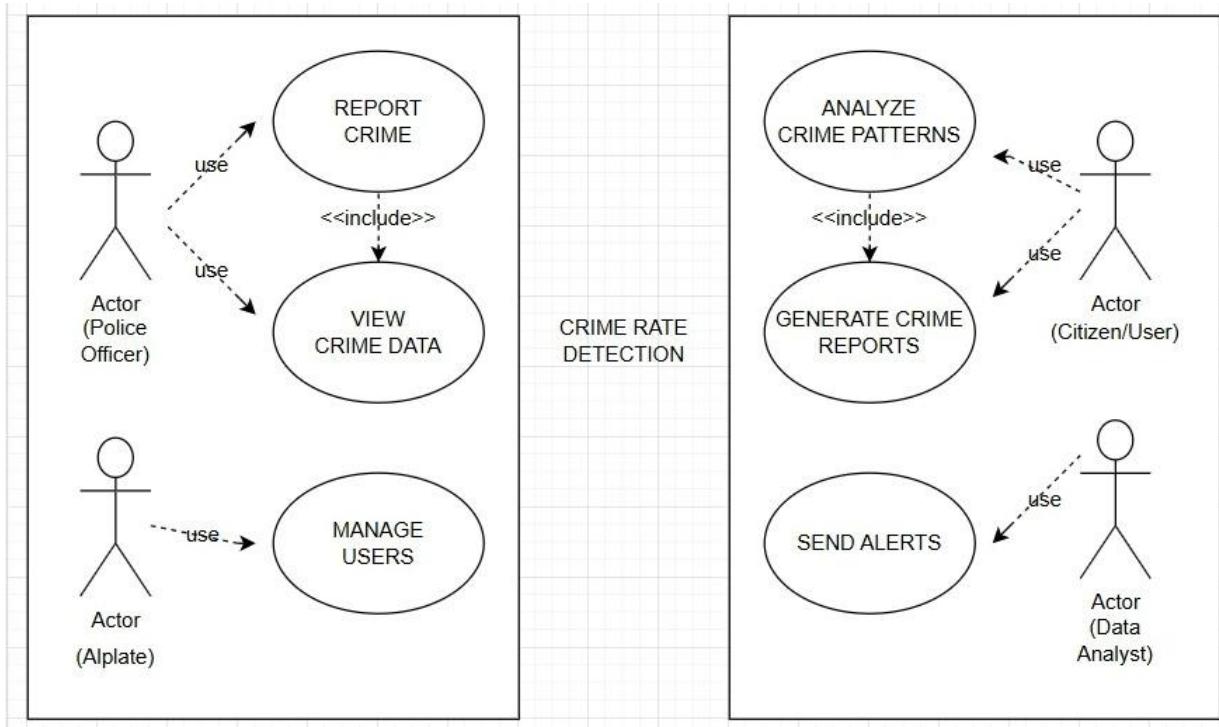
**EXP NO: 7**

## **DESIGNING USE-CASE AND ACTIVITY DIAGRAMS FOR PROJECT STRUCTURE**

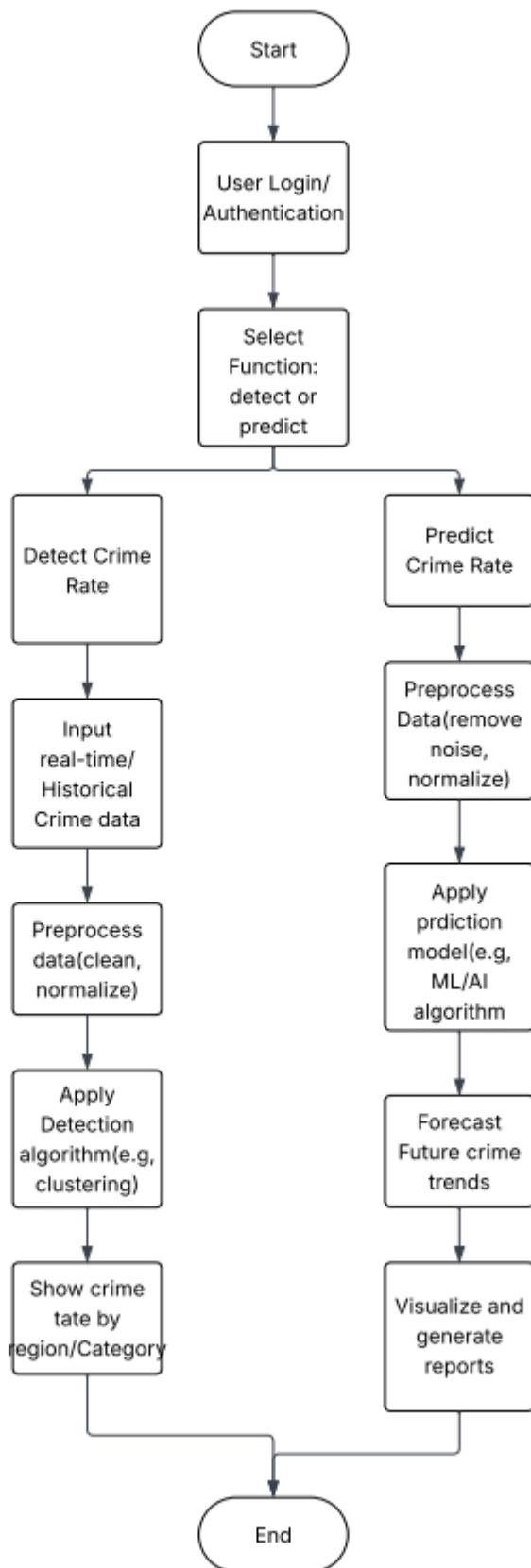
### **Aim:**

To Design an Use-Case Diagram and Activity Diagram for the given Project.

### **7A. Use-Case Diagram**



## 7B. Activity Diagram



**Result:**

The Use Case and Activity is designed Successfully for the Crime Rate Detector App

**EXP NO: 8**

## **TESTING – TEST PLANS AND TEST CASES**

### **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**

- Data Collection: Collect crime data from APIs, CSVs, public records, and databases.
- Data Preprocessing: Clean, normalize, and structure data (time, location, crime type).
- Machine Learning Prediction: Predict crime probability based on inputs like location and time.
- Risk Zone Identification: Highlight high-crime zones visually using maps or heatmaps.
- Reporting and Logs: Generate reports and logs for crime trends and predictions.

##### **2. Define User Interaction**

- A data engineer collecting data from a new source.
- A user submitting a location and date for crime prediction.
- A security analyst viewing the high-risk zone dashboard.

##### **3. Design Happy Path Test Cases**

- User enters valid input and receives a correct prediction.
- High-risk zones accurately visualized based on recent data.
- ML model processes cleaned and transformed data correctly.

##### **4. Design Error Path Test Cases**

- Missing input fields (e.g., location not provided).
- Invalid or corrupt data fails during preprocessing.
- System tries to predict with outdated or untrained model.

##### **5. Break Down Steps and Expected Results**

- **Step-by-step instructions** (e.g., "Enter location, select date, click 'Predict'").
- **Expected outcome** (e.g., "System displays crime risk level: Low/Medium/High").  
This ensures clarity for manual testers and consistency in automation scripts.

## 6. Use Clear Naming and IDs

- TC01 – Successful Crime Data Collection
  - TC05 – Crime Prediction with Valid Input
  - TC10 – Handle Missing Location Input
- This helps in mapping test cases to user stories or feature requirements.

## 7. Separate Test Suites

- Grouped test cases based on functionality (e.g., Data Collection ML Prediction Engine, Data Preprocessing).
- Improves organization and test execution flow in Azure DevOps.

## 8. Prioritize and Review

- Critical user actions are marked high-priority.
- Reviewed for completeness and traceability against feature requirements.

## 1. New test plan

The screenshot shows the 'New Test Plan' dialog in the Azure DevOps interface. The left sidebar shows the project navigation with 'Crime Rate Detector' selected. The main area displays the 'New Test Plan' form with the following fields:

- Name \***: Crime Rate Detector
- Area Path \***: Crime Rate Detector
- Iteration \***: Crime Rate Detector\Sprint 3 (with a date range from 05/05/2025 - 16/05/2025)

At the bottom right of the dialog are 'Create' and 'Cancel' buttons.

## 2. Test suite

The screenshot shows the Azure DevOps Test Plans interface. On the left, there's a sidebar with project navigation: Overview, Boards, Repos, Pipelines, Test Plans, Test plans, Progress report, Parameters, Configurations, Runs, and Artifacts. The 'Test plans' section is currently selected. In the main area, a test suite named 'Crime Rate Detector' is expanded, showing its sub-suites: 'TS01 – Crime Data Collect...' and 'New Suite'. A context menu is open over 'TS01 – Crime Data Collect...', listing options like 'New Suite', 'Assign configurations', 'Export', 'Open', 'Assign testers to run all tests', 'Rename', 'Delete', and 'Import test suites'. To the right, the specific test case details for 'TS01 – Crime Data Collection (ID: 77)' are displayed, including a table of three test cases: TC01-Source Connection Test, TC02-Fetch Data from All Sources, and TC03-Handle Duplicate Records.

## 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 Detector App – Test Plans

### USER STORIES

1. As a data engineer, I want to collect crime data from various sources so that I can create a comprehensive dataset.
2. As a data analyst, I want to extract time, location, and crime type from raw crime data so that I can structure it properly for accurate and efficient analysis.
3. As a data scientist, I want to encode categorical data and scale numerical features to make the data ML-ready.
4. As a developer, I want to clean inconsistent or missing values in the dataset so that I can ensure high-quality inputs for the machine learning model.
5. As a security analyst I want to identify high-risk zones So that I can allocate resources effectively.

### Test Suites

**Test Suite: TS01 – Crime Data Collection (ID: 77)**

**TC01 – Source Connection Test**

- **Action:**
  - Attempt connection to data source
  - Check for timeout or error

- **Expected Results:**
    - Connection established successfully
    - No timeout or error observed
  - **Type:** Happy Path
- 

### TC02 – Fetch Data from All Sources

- **Action:**
    - Trigger data retrieval
    - Log number of records retrieved
  - **Expected Results:**
    - Data retrieval process initiates
    - Record count is logged correctly
  - **Type:** Happy Path
- 

### TC03 – Handle Duplicate Records

- **Action:**
    - Load data from multiple sources
    - Check for duplicate crime IDs or timestamps
  - **Expected Results:**
    - All sources are successfully loaded
    - Duplicates flagged or removed
  - **Type:** Happy Path
- 

### Test Suite: TS02 – Data Structuring (ID: 81)

#### TC04 – Extract Timestamps

- **Action:**
    - Upload raw data
    - Extract timestamp field
  - **Expected Results:**
    - Data loads without errors
    - Dates standardized to YYYY-MM-DD format
  - **Type:** Error Path
- 

#### TC05 – Parse Location Data

- **Action:**
    - Process raw data
    - Extract location fields
  - **Expected Results:**
    - Data processed successfully
    - City, ZIP, and coordinates extracted correctly
  - **Type:** Happy Path
- 

#### TC06 – Classify Crime Types

- **Action:**
    - Analyze crime descriptions
    - Match against predefined categories
  - **Expected Results:**
    - Descriptions are readable and processed
    - Each record is assigned a crime type
  - **Type:** Happy Path
- 

#### Test Suite: TS03 – Feature Engineering (ID: 85)

##### TC07 – Encode Categorical Features

- **Action:**
    - Identify categorical fields
    - Apply Label/One-Hot encoding
  - **Expected Results:**
    - All categorical fields are found
    - Encoding completes without errors
  - **Type:** Happy Path
- 

##### TC08 – Scale Numerical Features

- **Action:**
  - Load dataset
  - Apply MinMax or StandardScaler
- **Expected Results:**
  - Dataset loads correctly
  - Values fall within expected scaled range
- **Type:** Happy Path

---

## **TC09 – Validate ML Input Schema**

- **Action:**
    - Run schema validator
    - Verify dataset
  - **Expected Results:**
    - Schema is read and applied
    - Dataset contains required fields with no nulls or incorrect formats
  - **Type:** Happy Path
- 

## **Test Suite: TS04 – Data Cleaning (ID: 89)**

### **TC10 – Check for Missing Values**

- **Action:**
    - Scan dataset
    - Identify null/empty fields
  - **Expected Results:**
    - Scan completes
    - Missing values highlighted or flagged
  - **Type:** Happy Path
- 

### **TC11 – Format Consistency Check**

- **Action:**
    - Load data
    - Validate formats (date, address, type)
  - **Expected Results:**
    - File loads successfully
    - All fields meet format specifications
  - **Type:** Happy Path
- 

### **TC12 – Remove Outliers**

- **Action:**
  - Apply outlier detection
  - Remove or normalize

- **Expected Results:**
    - Outliers detected correctly
    - Dataset updated with outliers handled
  - **Type:** Error Path
- 

## Test Suite: TS05 – High-Risk Zone Detection (ID: 93)

### TC13 – Calculate Zone Risk Score

- **Action:**
    - Aggregate crimes by location
    - Compute risk using frequency/severity
  - **Expected Results:**
    - Crime counts grouped correctly
    - Risk score reflects accurate level
  - **Type:** Error Path
- 

### TC14 – Generate Risk Heatmap

- **Action:**
    - Input data into visualization tool
    - Display heatmap
  - **Expected Results:**
    - Data loads to map tool
    - High-risk areas colored with appropriate intensity
  - **Type:** Happy Path
- 

### TC15 – List Top High-Risk Areas

- **Action:**
    - Sort areas by crime score
    - Display top 5 zones
  - **Expected Results:**
    - Areas ranked correctly
    - List includes top 5 zones with scores shown
  - **Type:** Happy Path
-

## Test Cases

The screenshot displays two test cases in the Azure DevOps Test Plan interface:

**Test Case 86: TC07-Encode Categorical Features**

- Steps:**
  - 1. Identify categorical fields - Expected result: All categorical fields are found
  - 2. Apply Label/One-Hot encoding - Expected result: Encoding completes without errors

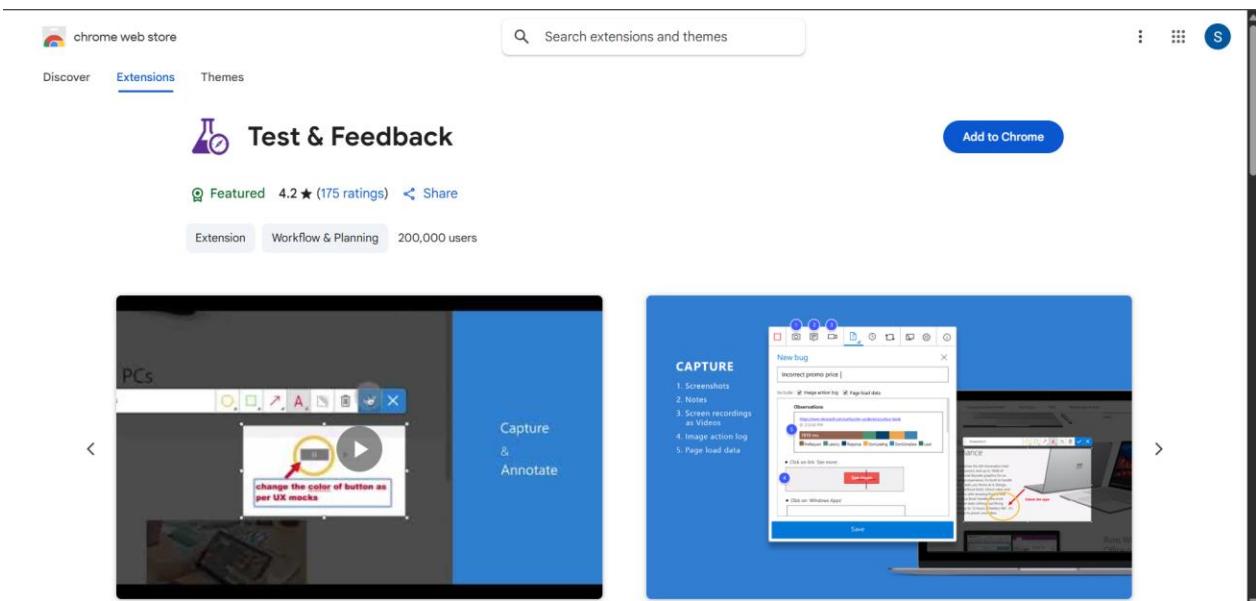
**Test Case 92: TC12-Remove Outliers**

- Steps:**
  - 1. Apply outlier detection - Expected result: Outliers detected correctly
  - 2. Remove or normalize - Expected result: Dataset updated with outliers handled

**Common UI Elements:**

- Left Sidebar:** Shows project navigation with sections like Crime Rate Detector, Overview, Boards, Reports, Pipelines, Tests, Programs, Parameters, Configuration, Runs, and Artifacts.
- Header:** Shows the URL dev.azure.com/2315011570509/Crime%20Rate%20Detector/\_testPlans/define?planId=75&suiteId=89.
- Top Right:** Includes a search bar, ribbon navigation (Save and Close, Follow, etc.), and a Help icon.
- Right Panel:** Contains sections for Deployment (status reporting), Development (linking to Azure Repos), and Related Work (adding links to other work items).

#### 4. Installation of test



#### Test and feedback

Showing it as an extension

A screenshot of the Azure DevOps interface, specifically the "Test Plans" section for a project named "Crime Rate Detector". The left sidebar shows "Test Plans" selected. The main area displays a "Test Suites" list with items like "TS01 – Crime Data Collection (ID: 77)", "TS02 – Data Structuring (3)", "TS03 – Feature Engineering (3)", "TS04 – Data Cleaning (3)", and "TS05 – High-Risk Zone De...". A modal window for "TS01 – Crime Data Collection" is open, showing "Test Cases (3 items)" with entries for "TC01-Source Connection Test", "TC02-Fetch Data from All Sources", and "TC03-Handle Duplicate Records". On the right, an "Extensions" sidebar is visible, listing "McAfee® WebAdvisor" with "Full access" and a link to "Manage extensions".

## 5. Running the test cases

The screenshot shows the Azure DevOps Test Plans interface. On the left, the sidebar navigation includes Crime Rate Detector, Overview, Boards, Repos, Pipelines, Test Suites, Test plans, Progress report, Parameters, Configurations, Runs, Artifacts, and Project settings. The 'Test plans' section is currently selected.

In the main area, the 'Crime Rate Detector' project is selected. A summary bar indicates 'May 5 - May 16' and '40% run, 83% passed'. Below this is a 'Test Suites' section with a dropdown menu and a 'Filter suites by name' input field. A list of suites is shown, with 'TS01 – Crime Data Collection (ID: 77)' selected.

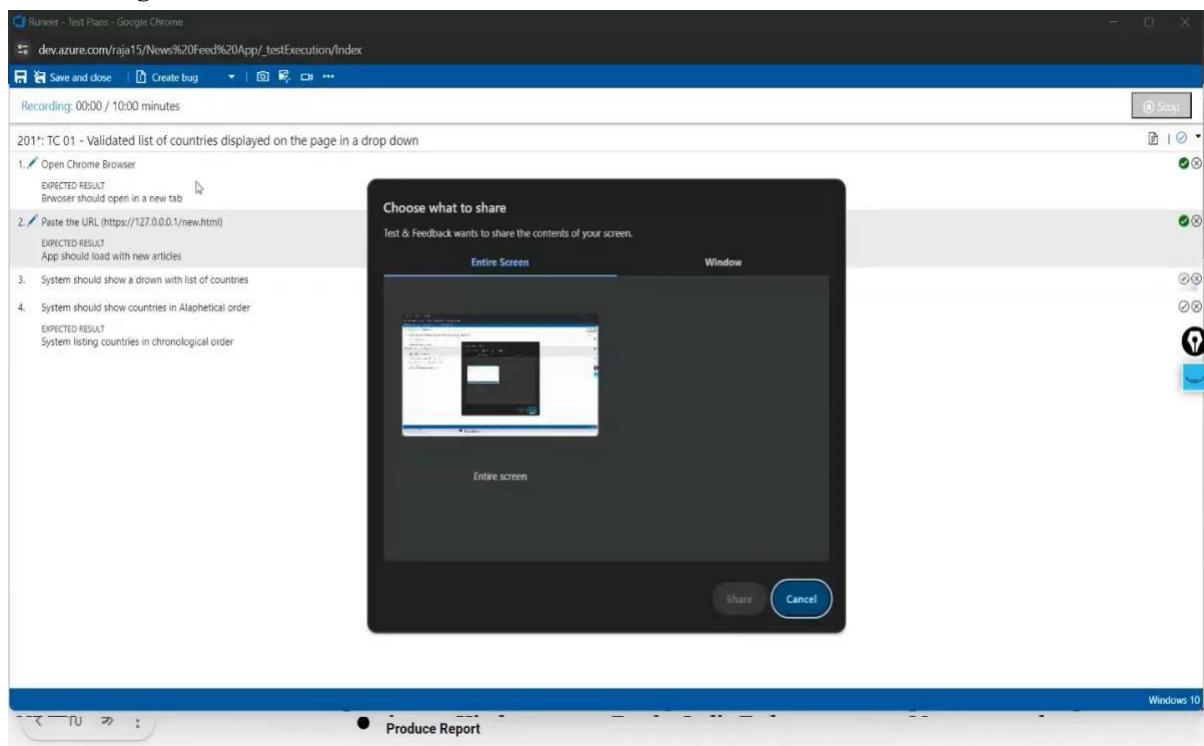
The 'TS01 – Crime Data Collection (ID: 77)' page displays a 'Test Points (3 items)' table:

Title	Outcome	Order	Test Case Id
TC01-Source Connection Test	Failed	1	78
TC02-Fetch Data from All Sources	Passed	2	79
TC03-Handle Duplicate Records	Passed	3	80

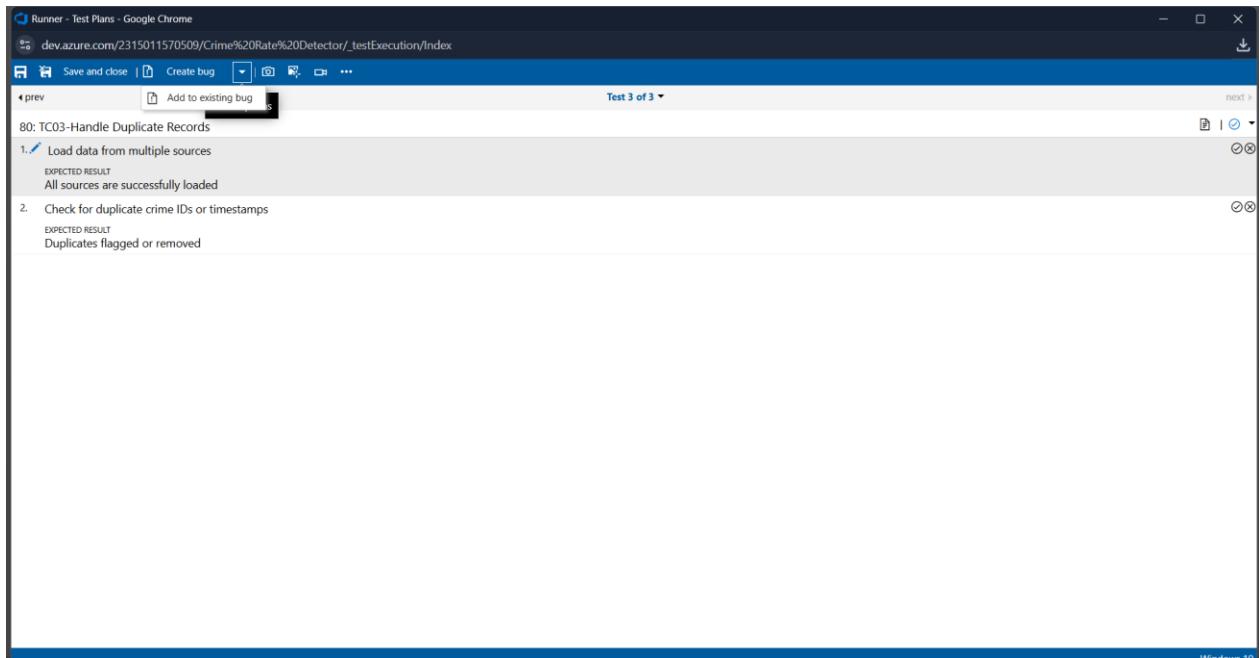
Below this, a 'Test Execution' window is open, showing 'Test 1 of 3' for '78: TC01-Source Connection Test'. The steps are:

1. Attempt connection to data source  
EXPECTED RESULT  
Connection established
2. Check for timeout or error  
EXPECTED RESULT  
No timeout or error observed

## 6.Recording the test case



## 7.Creating the bug



Runner - Test Plans - Google Chrome  
dev.azure.com/2315011570509/Crime%20Rate%20Detector/\_testExecution/Index

Save and close Create bug ...

prev Test 3 of 3 next

80: TC03-1  
1. Load EXPECT All sources  
2. Check EXPECT Duplicate records

NEW BUG + TB01-Duplicate crime records not flagged or removed after loading multiple sources

Unassigned 0 comments Add tag

State: New Area: Crime Rate Detector  
Reason: New Iteration: Crime Rate Detector\Sprint 3

Repro Steps  
20/05/2025 11:44 Bug filed on "TC03-Handle Duplicate Records"

Step no.	Result	Title
1.	None	Load data from multiple sources Expected Result All sources are successfully loaded
2.	None	Check for duplicate crime IDs or timestamps Expected Result Duplicates flagged or removed

Planning  
Resolved Reason: None  
Story Points: 1  
Priority: 2  
Severity: 3 - Medium  
Activity: Ram simon

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  
Add an existing work item as a parent

Windows 10

This screenshot shows a detailed view of a bug work item in the Azure DevOps Test Plan. The bug is titled 'TB01-Duplicate crime records not flagged or removed after loading multiple sources'. It is categorized under 'CrimeType Detector' and is currently 'Unassigned'. The 'Repro Steps' section contains two steps: one for loading data from multiple sources and another for checking for duplicate crime IDs or timestamps. The 'Planning' section includes fields for priority, severity, and activity. The 'Deployment' and 'Development' sections provide links to external resources and branches. The 'Related Work' section allows for linking to other work items. The bottom of the screen shows the Windows 10 taskbar.

Runner - Test Plans - Google Chrome  
dev.azure.com/2315011570509/Crime%20Rate%20Detector/\_testExecution/Index

Save & Close ...

NEW BUG + TB01-Duplicate crime records not flagged or removed after loading multiple sources

Unassigned 0 comments Add tag

State: New Area: Crime Rate Detector  
Reason: New Iteration: Crime Rate Detector\Sprint 3

System Info

Browser - Name	Google Chrome 136
Browser - Language	en-US
Browser - Height	816
Browser - Width	1536
Browser - User agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
Operating system - Name	Windows NT 10.0; Win64; x64
Operating system - Architecture	x86_64
Operating system - Processor model	13th Gen Intel(R) Core(TM) i7-13650HX
Operating system - Number of processors	20
Memory - Available	8027467776
Memory - Capacity	16873545728
Display - Pixels per inch (X axis)	120
Display - Pixels per inch (Y axis)	120
Display - Device pixel ratio	1.25

Discussion  
Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.

This screenshot shows the 'System Info' table for the bug item. The table lists various system configuration details such as browser type (Google Chrome 136), operating system (Windows NT 10.0; Win64; x64), processor (13th Gen Intel(R) Core(TM) i7-13650HX), and memory capacity (16873545728). The 'Found in Build' and 'Integrated in Build' status indicators are visible on the right side of the table.

## 8. Test case results

The screenshot shows the Azure DevOps interface for a project named "Crime Rate Detector". The left sidebar has "Test Plans" selected under "Crimes Rate Detector". The main area displays a "Test Suites" list with "TS01 – Crime Data Collection (ID:7)" expanded. Under it, three test points are listed: "TC01-Source Connection Test" (Passed), "TC02-Fetch Data from All Sources" (Paused), and "TC03-Handle Duplicate Records" (Failed). A modal window titled "TC01-Source Connection Test" shows the "Test Case Results" tab with a table of results:

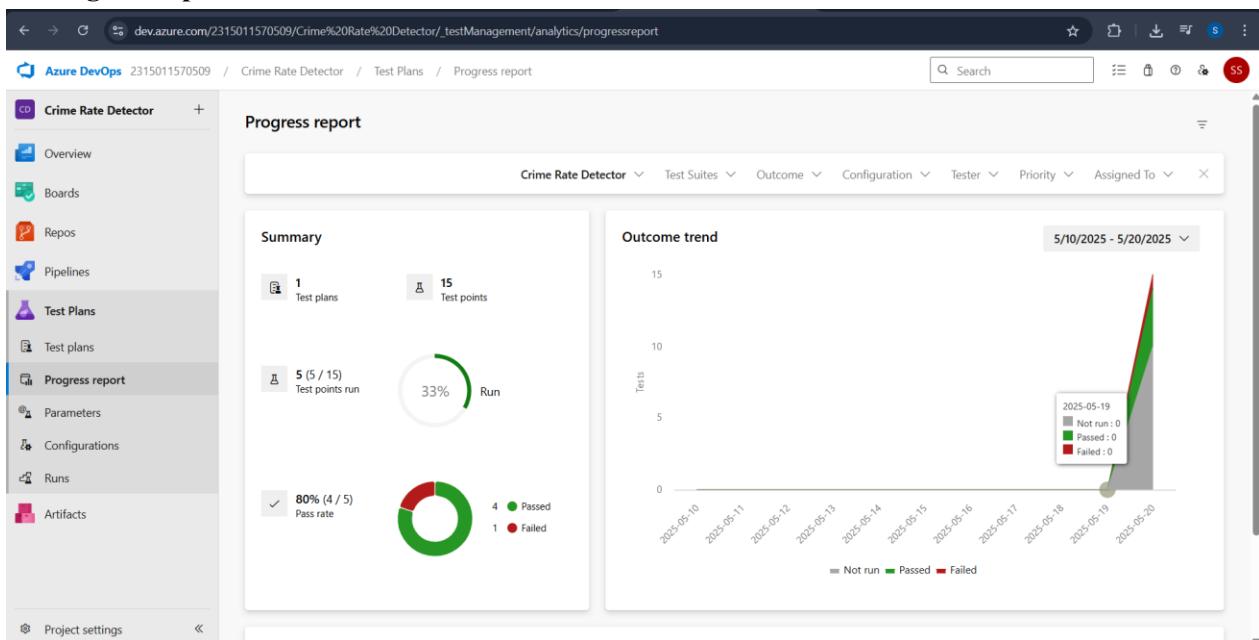
Outcome	TimeSta...	Configuration	Run by	Tester	Test Plan
Passed	11m ago	Windows 10	Sonia J S	Sonia J S	Crime F...
Paused	30m ago	Windows 10	Sonia J S	Sonia J S	Crime F...
Failed	4h ago	Windows 10	Sonia J S	Sonia J S	Crime F...

## 9. Test report summary

The screenshot shows the Azure DevOps interface for the same project. The left sidebar has "Work items" selected. A specific work item, "BUG 97 TB01-Duplicate crime records not flagged or removed after loading multiple sources", is displayed in detail. The work item form includes fields for State (New), Reason (New), Repro Step (Active, Resolved, Closed), Planning (Resolved Reason, Story Points, Priority, Severity, 3 - Medium, Activity), Deployment (Deployment status reporting), Development (Add link, Link an Azure Repos commit, pull request or branch), and Effort (Hours) (Original Estimate). The work item was updated by Sonia J S 23m ago.

- Assigning bug to the developer and changing state

## 10. Progress report



## 11. Changing the test template

The screenshot shows the 'All processes' section of the Azure DevOps 'Organization Settings' page. The sidebar on the left lists 'Organization Settings' (2315011570509), 'General' (Overview, Projects, Users, Billing, Global notifications, Usage, Extensions, Microsoft Entra), 'Security' (Security overview, Policies, Permissions), and 'Boards' (Process). The main content area displays a table of available process templates:

Name	Description	Team projects
Basic (default)	This template is flexible for any process and great for teams getting started with Az...	0
Agile	... This template is flexible and will work great for most teams using Agile planning me...	1
Scrum	This template is for teams who follow the Scrum framework.	0
CMMI	This template is for more formal projects requiring a framework for process improv...	0

The screenshot shows the Azure DevOps 'Process' settings page for a specific organization setting (2315011570509). The left sidebar includes sections for General, Security, and Boards. The main content area displays the 'All processes > Agile' section, listing various work item types with their descriptions:

Name	Description
Bug	Describes a divergence between required and actual behavior, and tracks the work done to correct the defect and verify the correct...
Epic	Epics help teams effectively manage and groom their product backlog
Feature	Tracks a feature that will be released with the product
Issue	Tracks an obstacle to progress.
Task	Tracks work that needs to be done.
Test Case	Server-side data for a set of steps to be tested.
Test Plan	Tracks test activities for a specific milestone or release.
Test Suite	Tracks test activities for a specific feature, requirement, or user story.
User Story	Tracks an activity the user will be able to perform with the product.

### 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**

## **CI/CD PIPELINES IN AZURE**

### **AIM**

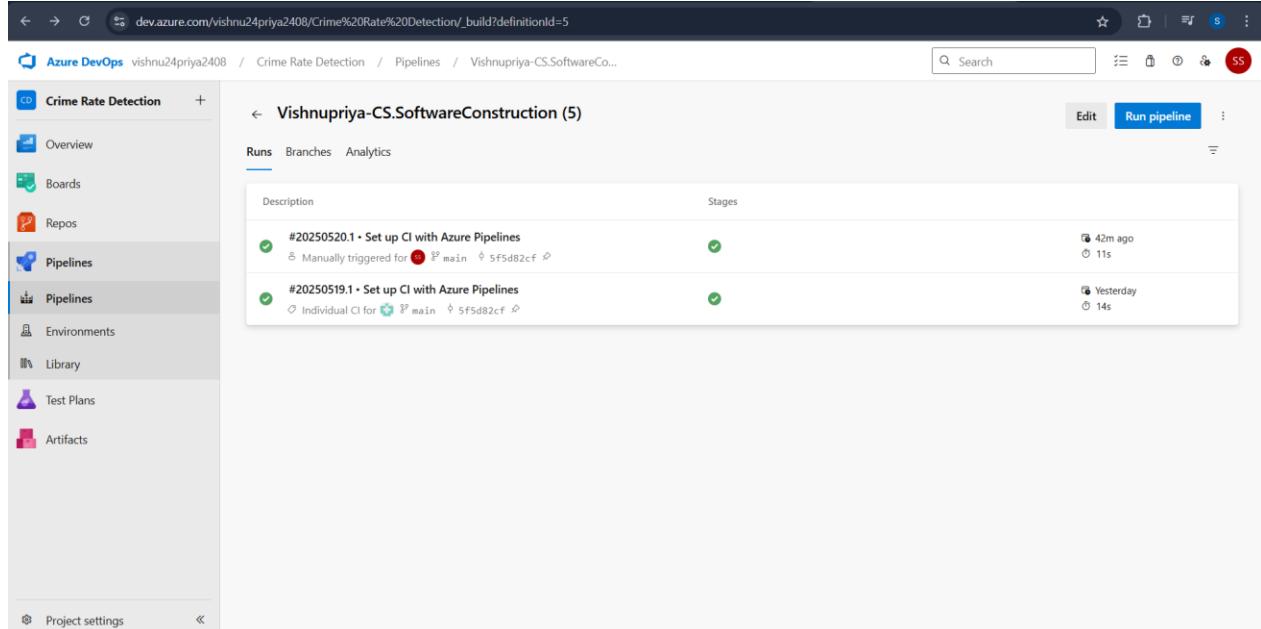
To implement a Continuous Integration and Continuous Deployment (CI/CD) pipeline in Azure DevOps for automating the build, testing, and deployment process of the Student Management System, ensuring faster delivery and improved software quality.

### **PROCEDURE:**

#### **Steps to Create and implement pipelines in Azure:**

1. Sign in to Azure DevOps and Navigate to Your Project Log in to dev.azure.com, select your organization, and open the project where your Student Management System code resides.
2. Connect a Code Repository (Azure Repos or GitHub) Ensure your application code is stored in a Git-based repository such as Azure Repos or GitHub. This will be the source for triggering builds and deployments in your pipeline.
3. Create a New Pipeline Go to the Pipelines section on the left panel and click “Create Pipeline”. Choose your source (e.g., Azure Repos Git or GitHub), and then select the repository containing your project code.
4. Choose the Pipeline Configuration You can select either the YAML-based pipeline (recommended for version control and automation) or the Classic Editor for a GUI-based setup. If using YAML, Azure DevOps will suggest a template or allow you to define your own.
5. Define Build Stage (CI - ContinuousIntegration) from YAML file.
6. Install dependencies (e.g., npm install, dotnet restore).
7. Build the application (dotnet build, npm run build).
8. Run unit tests(dotnet test, npm test).

9. Publish build artifacts to be used in the release stage.
10. Save and Run the Pipeline for the First Time Save the YAML or build definition and click “Run”. Azure will fetch the latest code and execute the defined build and test stages.
11. Configure Continuous Deployment (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).
12. Configure the CD stage with deployment tasks such as deploying to Azure App Service, running database migrations or scripts, and restarting services using the Azure App Service Deploy task linked to your subscription and app details.
13. Set Triggers and Approvals Enable continuous deployment trigger so the release pipeline runs automatically after a successful build. For production environments, configure pre-deployment approvals to ensure manual verification before release.
14. Monitor Pipelines and Manage Logs View all pipeline runs under the Runs section. Check logs for build/test/deploy stages to debug any errors. You can also integrate email alerts or Microsoft Teams notifications for build failures.
15. Review and Maintain Pipelines Regularly update your pipeline tasks or YAML configurations as your application grows. Ensure pipeline runs are clean and artifacts are stored securely. Integrate quality gates and code coverage policies to maintain code quality



The screenshot shows the Azure DevOps Pipelines Runs page for the project "Crime Rate Detection". The left sidebar has "Pipelines" selected. The main area shows the "Runs" tab for the pipeline "Vishnupriya-CS.SoftwareConstruction (5)". Two runs are listed:

Description	Stages	Time	Duration
#20250520.1 • Set up CI with Azure Pipelines ↳ Manually triggered for main ↴ 5f5d82cf ↴	Green checkmark	42m ago	11s
#20250519.1 • Set up CI with Azure Pipelines ↳ Individual CI for main ↴ 5f5d82cf ↴	Green checkmark	Yesterday	14s

The screenshot shows the Azure DevOps Pipelines interface for the project "Crime Rate Detection". The pipeline run "#20250520.1 • Set up CI with Azure Pipelines" has completed successfully. Key details include:

- Run status:** Success
- Run ID:** #20250520.1
- Run by:** Vishnupriya-CS.SoftwareConstruction (5)
- Time started and elapsed:** Today at 5:58 PM, 11s
- Related:** 0 work items
- Tests and coverage:** 1 published

The "Jobs" section shows one job named "Job" which was successful and took 5s.

### Result:

Thus the pipelines for the given project “Crime Rate Detector” has been executed successfully

**EXP NO:** 10

# GITHUB: PROJECT STRUCTURE & NAMING CONVENTIONS

## 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 Music Playlist Batch Creator project.

## GitHub Project Structure

The screenshot shows a GitHub repository named "Software-Construction-Lab". The repository has 2 commits, 1 branch, and 0 tags. It contains files: README.md, index.html, and style.css. The README file content is: file:///C:/Users/SONIA/OneDrive/Desktop/SC\_CRD/index.html. The repository has 0 stars, 1 watching, and 0 forks. It also shows sections for Releases, Packages, and Languages.

## 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.