Capstone Project – Documentation

Bug Tracking System (MVC - Single Controller)

Project Overview Create a basic bug tracker where users can log software bugs/issues. The system allows tracking bug status, severity, and assignee in a clean interface.

Table of Contents

   Bug Table

Field Name	Type
Id	int (PK)
Title	string
Description	string
Severity	string (Low, Medium, High)
Status	string (Open, In Progress, Closed)
Assignee	string
ReportedDate	DateTime

   BugsController.cs
   Handles:
   Index() – Show list of all bugs
   Create() – Submit a new bug
   Edit(id) – Update bug details (e.g., status/severi
   Details(id) – View bug details
   Delete(id) – Delete bug record

5. Views

Views/Bugs/
├── Index.cshtml
├── Create.cshtml
├── Edit.cshtml
├── Details.cshtml
└── Delete.cshtml

6. Folder Structure

BugTrackingSystem/
├── Controllers/
│   └── BugsController.cs
├── Models/
│   └── Bug.cs
├── Views/
│   └── Bugs/
├── Data/
│   └── ApplicationDbContext.cs

7. Deployment

Push to GitHub
Deploy to Azure App Service
Use Azure SQL Database
Configure appsettings.json with cloud DB connection

8. Documentation

Setup and usage instructions
Screenshot of the bug list and form
Deployment steps

8 . Deployment

8.1 Setup and Usage Instructions :

Prerequisites

Make sure the following are installed:

- Visual Studio 2022 + with **ASP.NET and web development** workload

- SQL Server (Express or full version)

- Git (for version control)

- Entity Framework Core packages installed via NuGet:

  o Microsoft.EntityFrameworkCore.SqlServer

Steps to Follow :

Step-1 : Open the Visual Studio 2022 in the system.

Click on the Create New Project, a page will open .

Right side in the top a search bar will be visible search for ASP.NET Core

MVC, It will come select that one and type next.
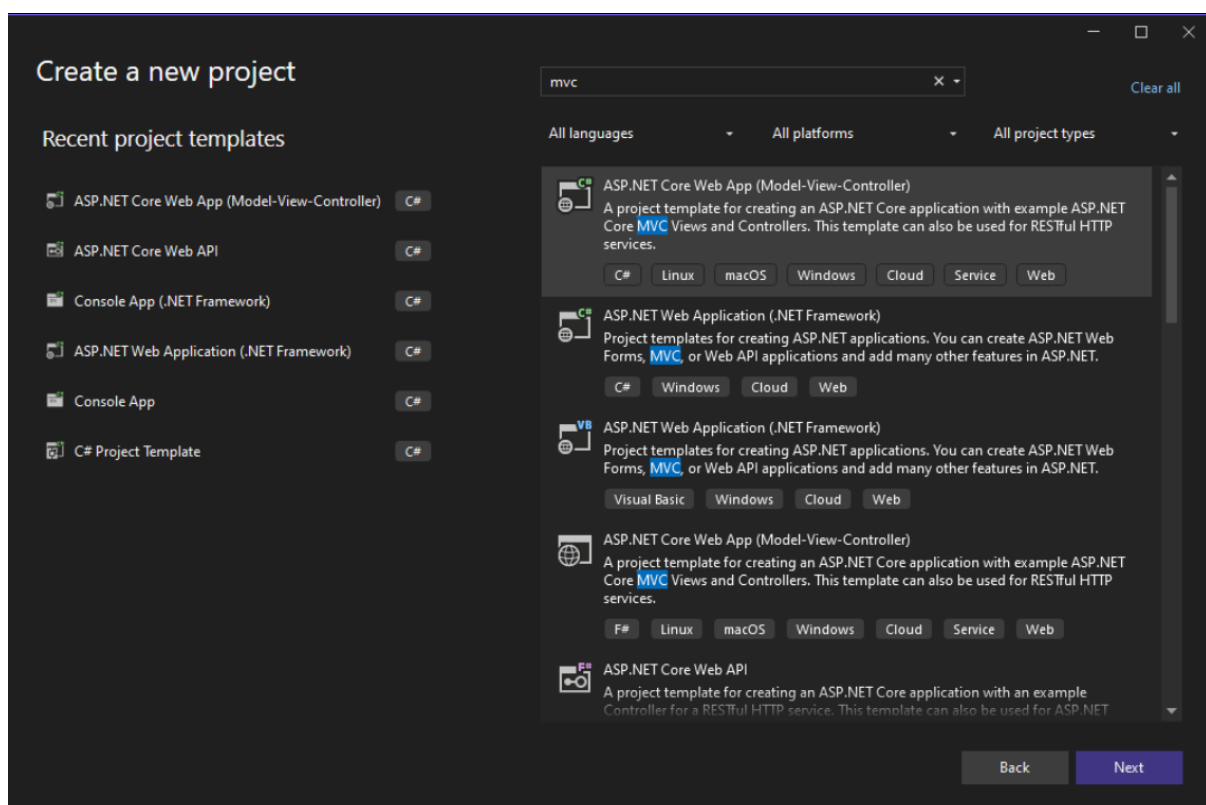
In that page give the name for Project as PROJECTS, then click next.

Additional information page will be visible there uncheck the box Configure

For HTTPS and then click Create.

Step-2 : On the Models folders give the right click and add a class with name Bug.cs
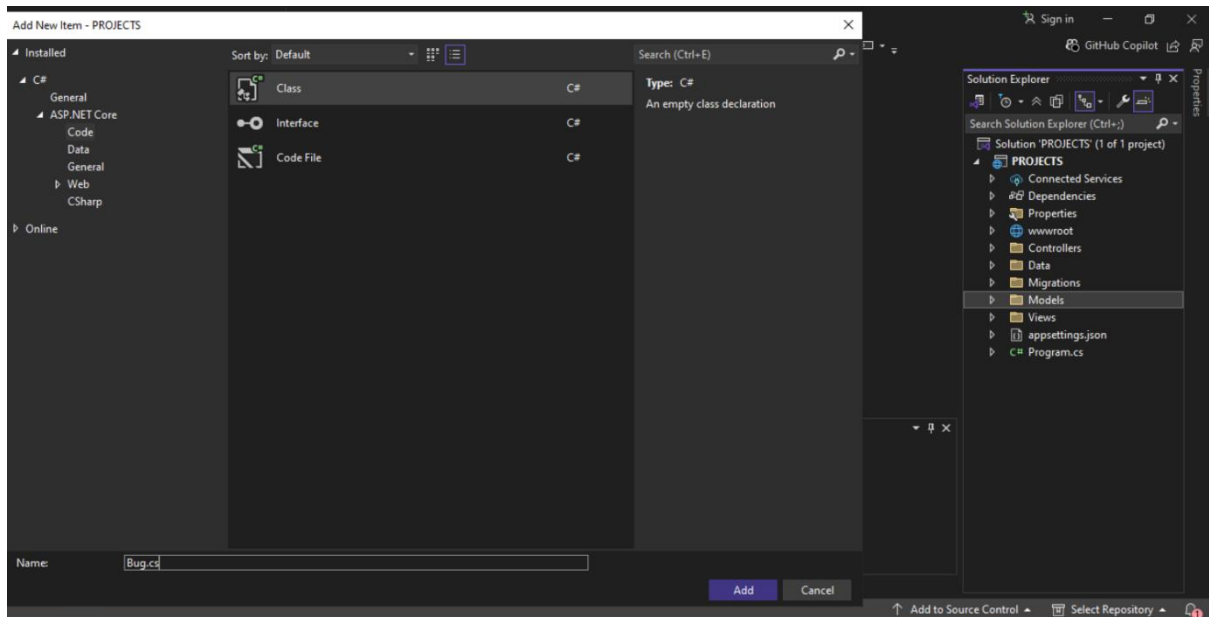
Click on Add, Bug.cs class will be created.

Step-3 : Add fields into the Bug.cs class
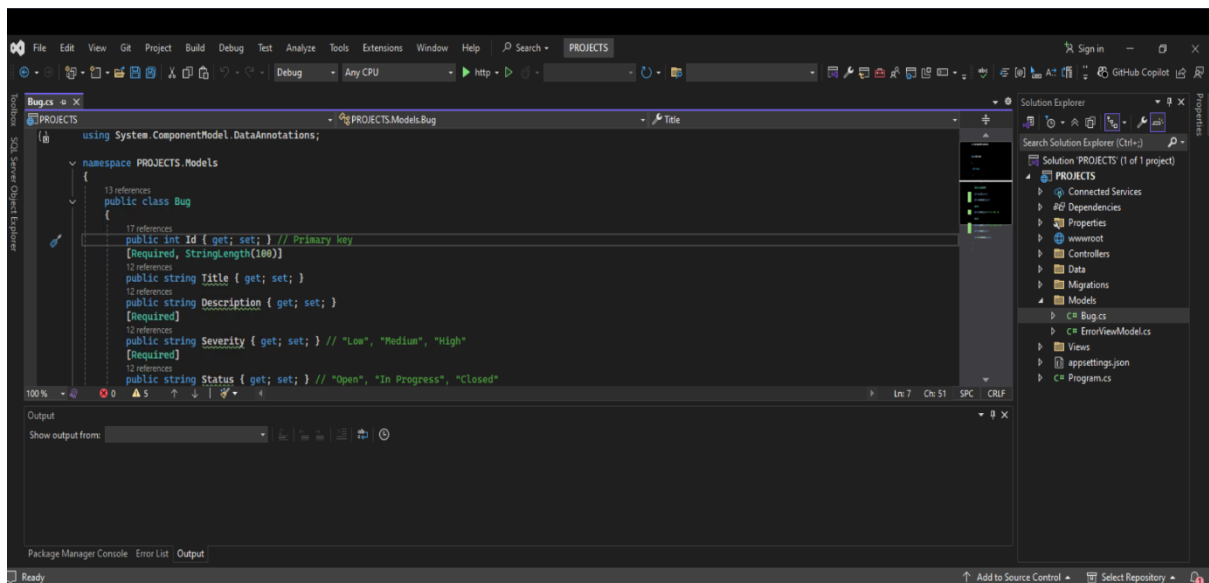
Step-4 : On controllers folder right click on it and Add , if you go to add you will

Find controller option.
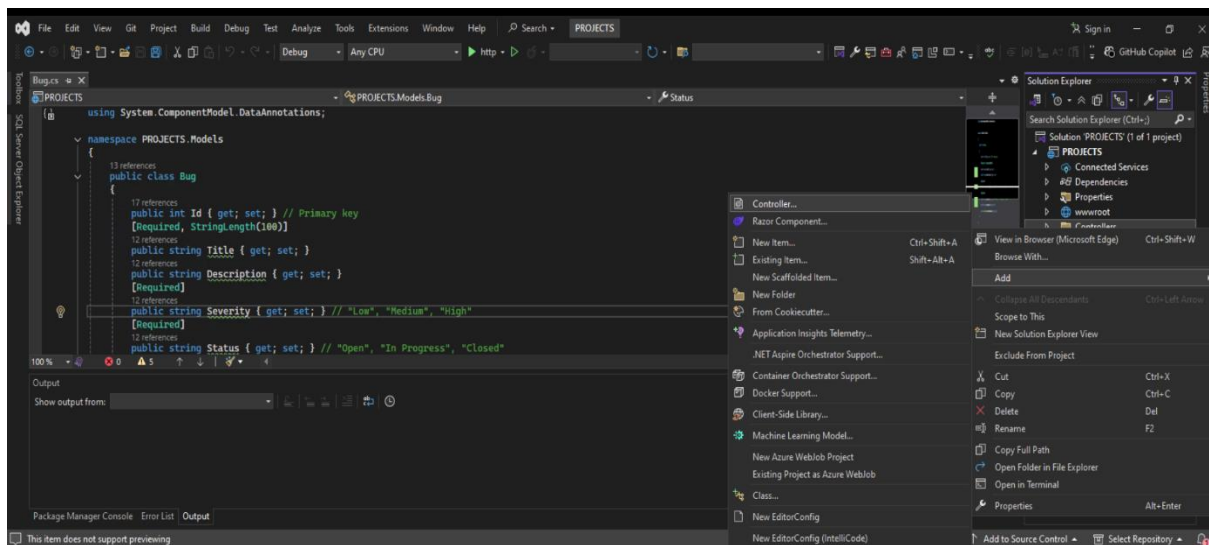
You will find a Add new Scaffolded Item Page select the MVC Controller with views Using Entity Framework then add.
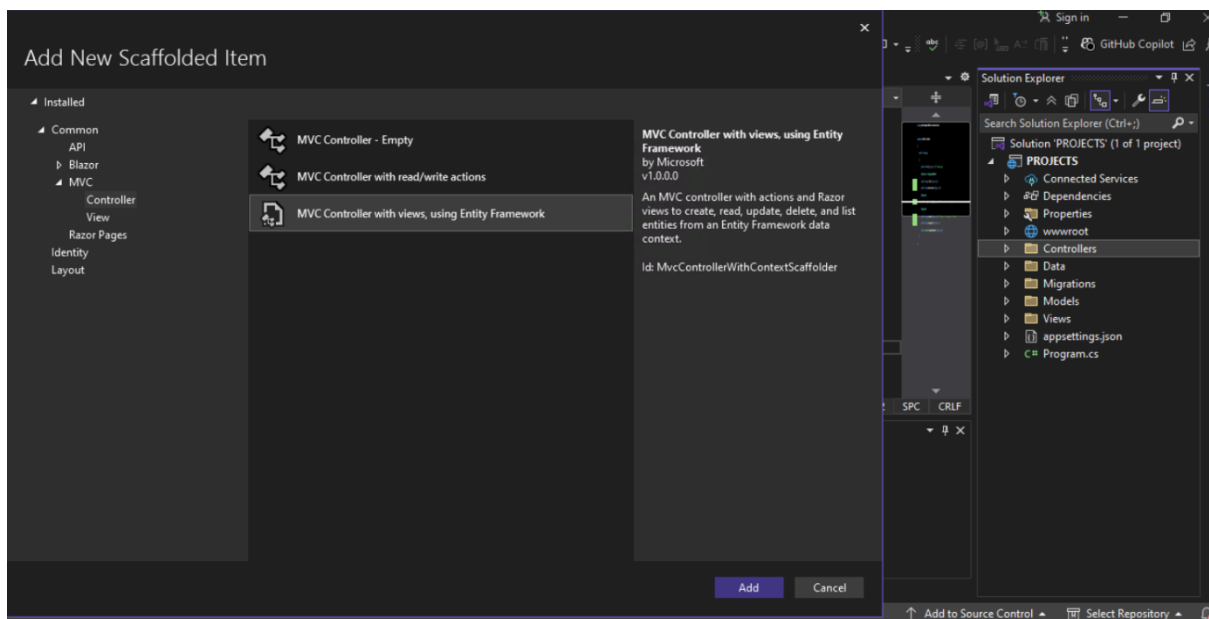
A new page will be visible with ADD MVC Controller with views , using Entity Framework.
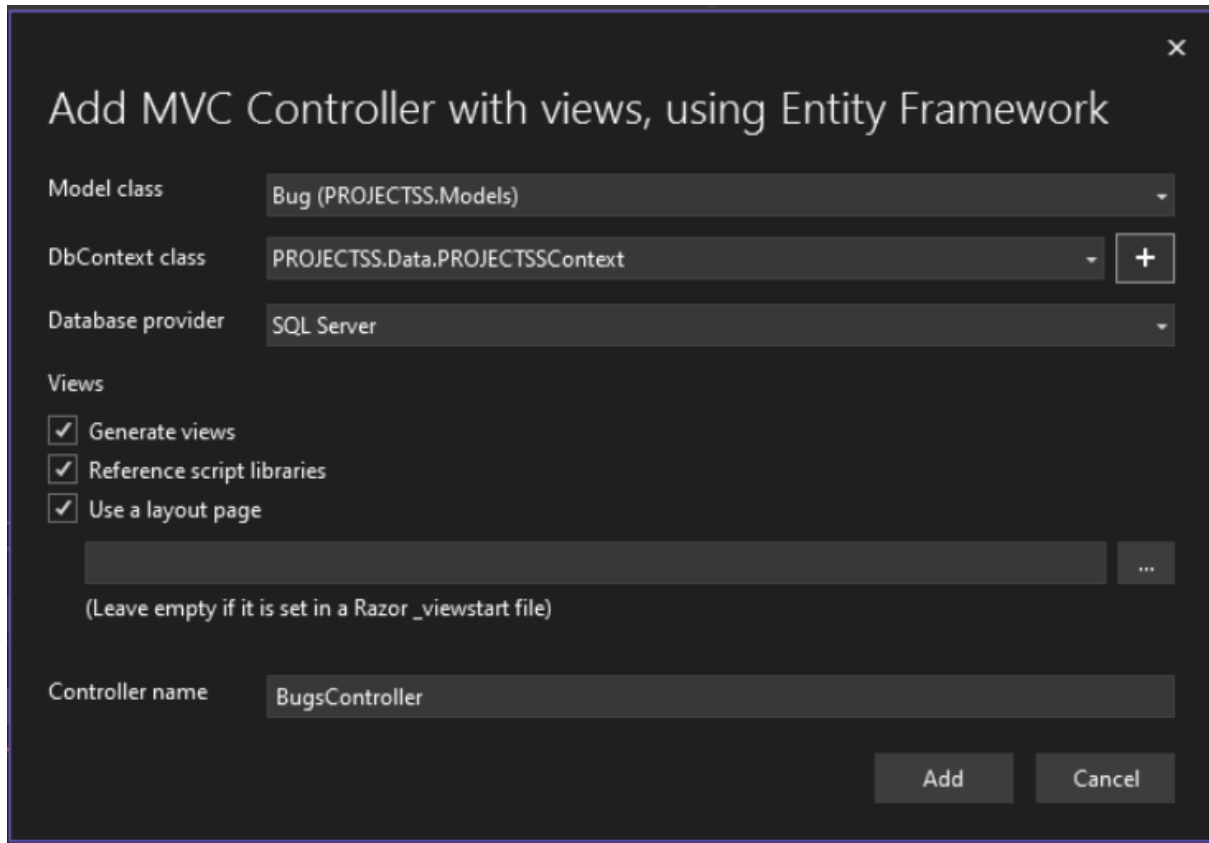
Select the Model class with Bug (PROJECTS.Models)

Add DbContext class with name PROJECTS.Data.PROJECTSContext.

Automatically SQL will be enables in the Database Provider.

The Controller name will be BugsController by default.

Then click on Add.

A new controller will be created with name BugsController.

Step-5 : In package Manager Console run these following commands

- Add-Migration InitialCreate
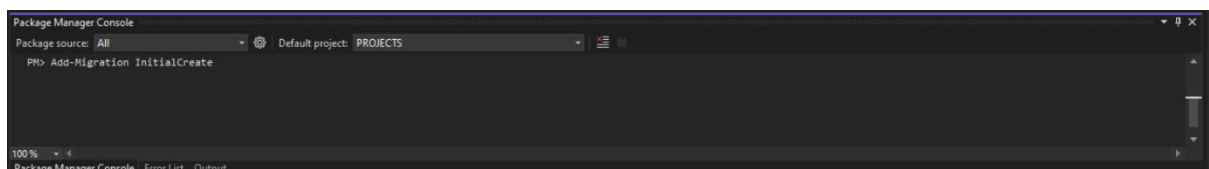
- Update-Database
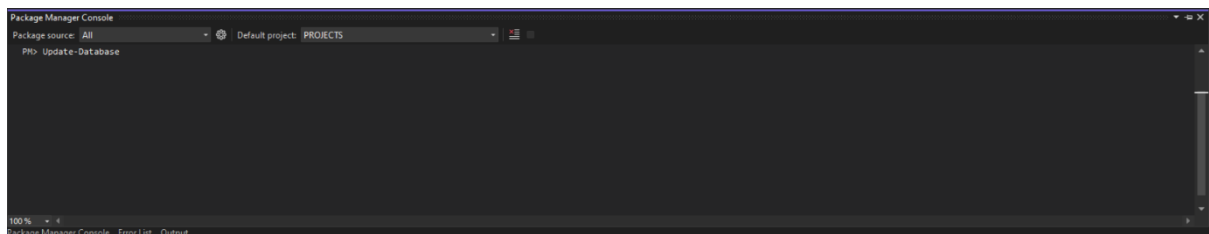
Step-6 : Run the controller program a simple web page will be created with homepage.
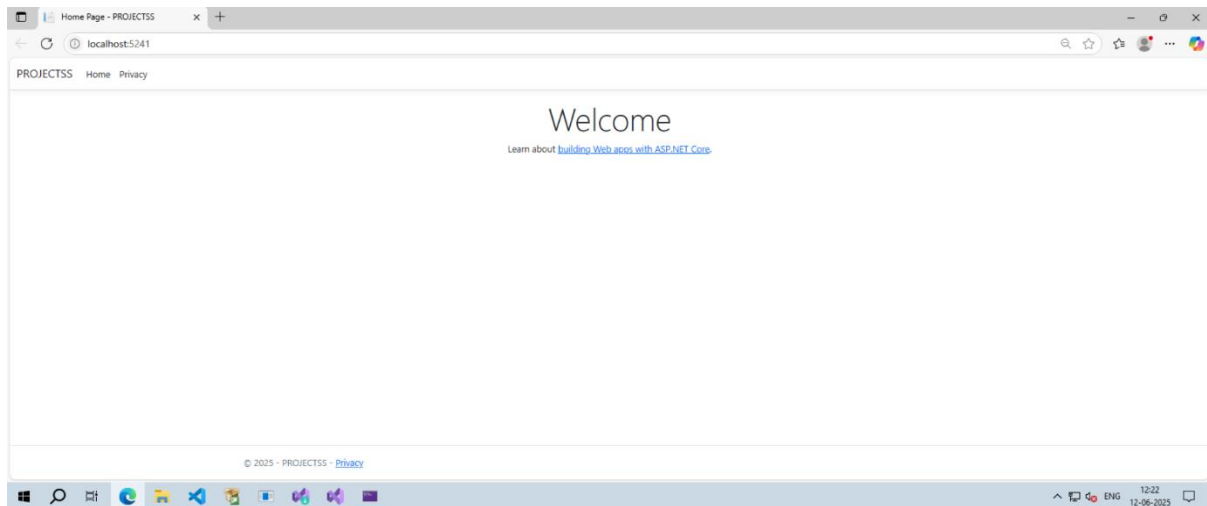
Step-7 : Go to localhost:xxxx/Bugs/Index
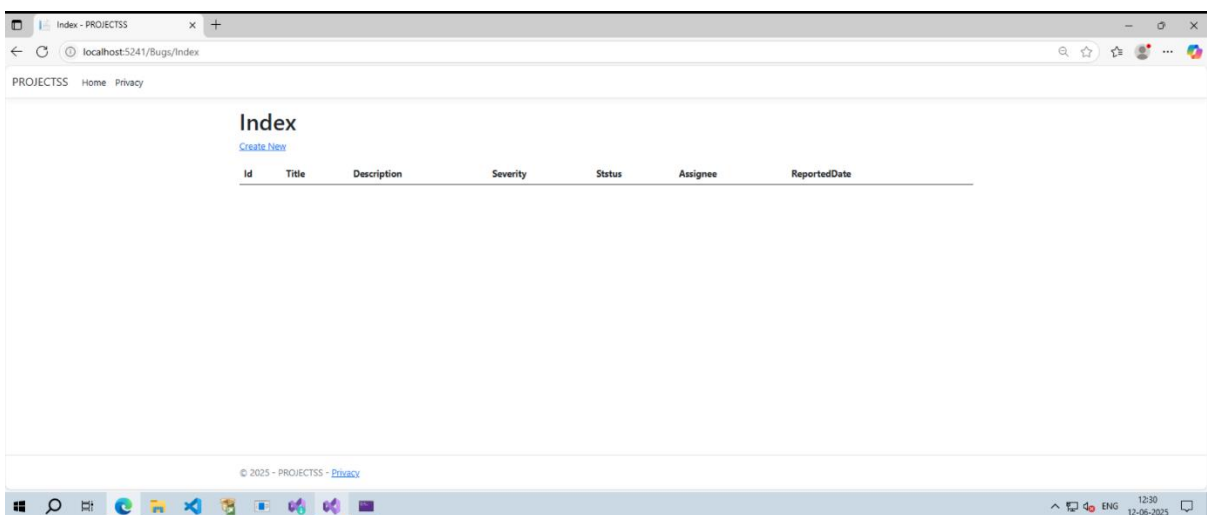
Step-8 : Add the Style part for

Views/Bugs/Index.cshtml

Views/Bugs/Create.cshtml

Views/Bugs/Edit.cshtml

Views/Bugs/Details.cshtml

Views/Bugs/Delete.cshtml with any one of HTML, CSS, Razor Views, Bootstrap.

Step-9 : Add style part for Views/Home/Index.cshtml

Then the homepage will look like
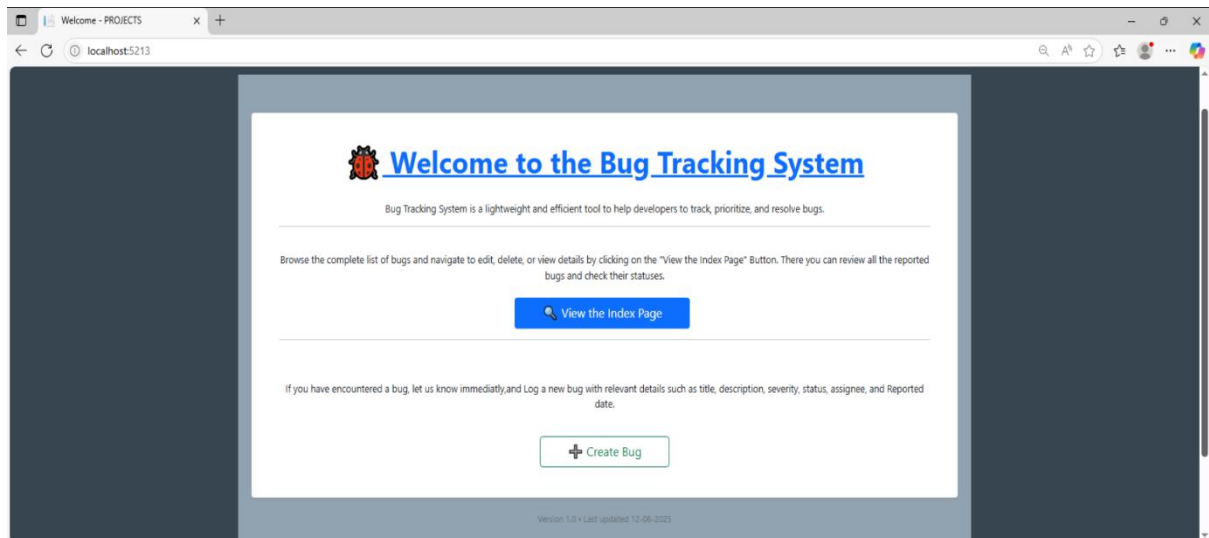
Usage Instructions :

Step-1 : **Index Page**

After running the application in real life if we login into the page A welcome home screen will be visible, after clicking on "View the Index Page"/by giving "localhost:xxxx/Bugs/Index" then it will navigate to Index page.

Here we can see the index page, it is having List of Bugs, we can create new bug, or we can delete the particular bug, or we can see the details of bugs individually, we can also edit the bugs.

Step-2 : **Create**

If we want to Create a new bug then Click On the "Create Bug" button, It will navigate to the localhost:xxxx/Bugs/Create page .There we can see the Report a New Bug page.

There we can see the Title, Description, Severity, Status, Assignee, ReportedDate as Id is by default.
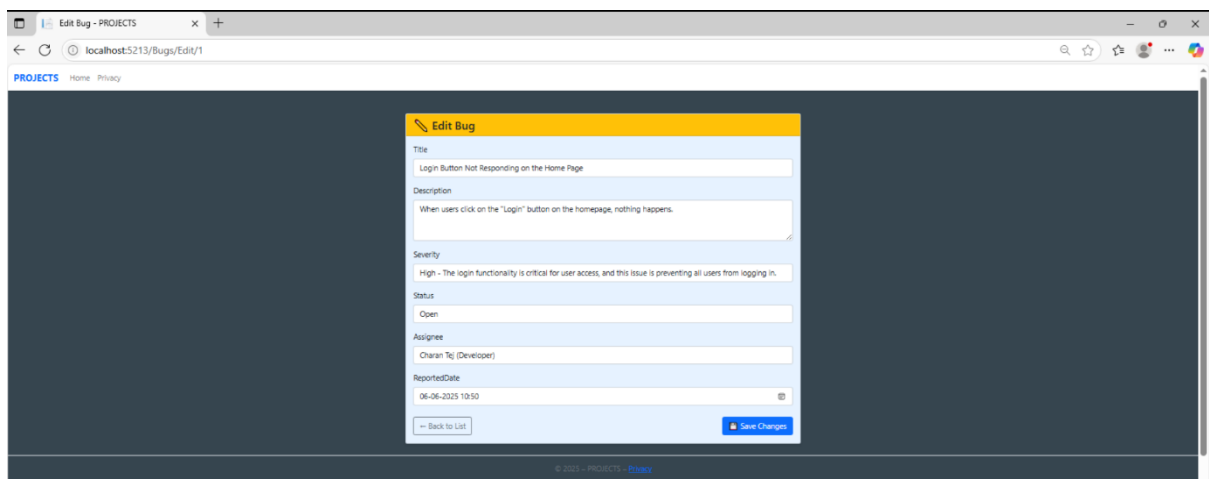
We have to fill the details and then click on "Create".

Step-3 : **Edit**

If we want to edit the particular entry we can click on the "Edit" button or we can give "localhost:xxxx/Bugs/Edit" it will navigate to the edit page.

There we can edit the details which we want to change, change the details and click on "Save Changes" button , changes will be saved.

Step-4 : **Details**

In Index page we can see the Details button, So if we want to know about particular bug information then click on the particular bug details ,it will navigate as shown below.
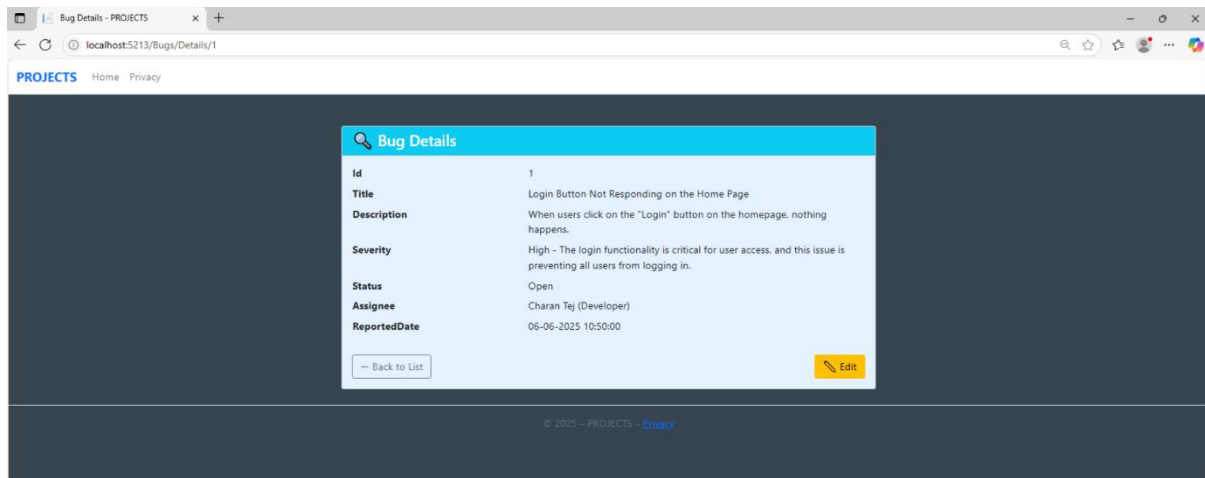

Here the navigation is localhost:xxxx/Bugs/Details/1.

Here we can see the individual bug details, like Id, Title of the bug, Description, Severity, Status, Assignee, ReportedDate.

If we don't want any detail we can edit.

Step-5 : **Delete**

So if we don't want any bug record then we can select the particular bug delete button, or we can give manually like "localhost:xxxx/Bugs/Delete/x then it will navigate to that page.
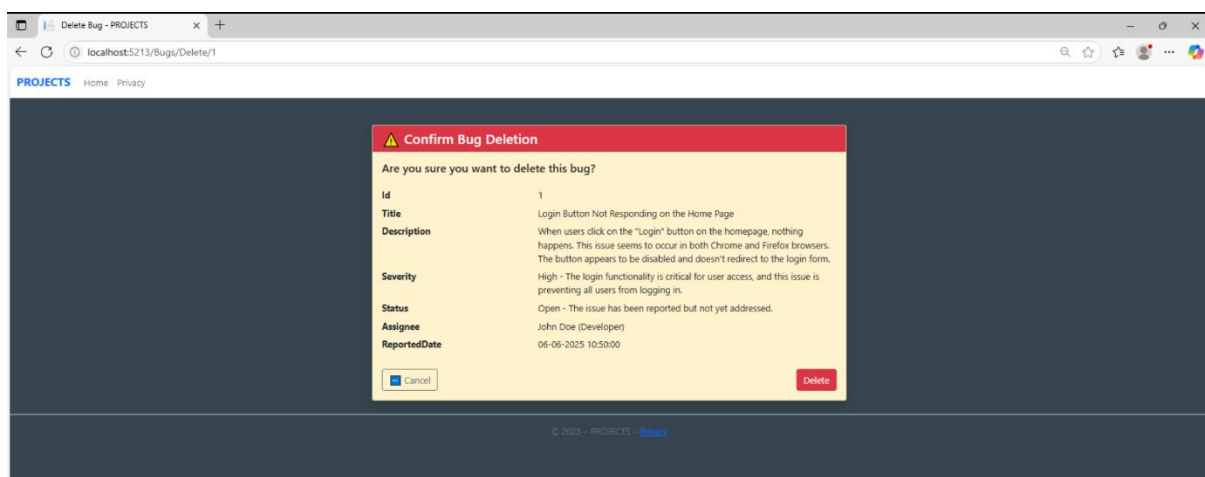
It is showing a warning here that we want to delete this bug or not.

If we click on delete the particular bug will be deleted.

8.2 Screenshot of the bug list and form:

Bug list:

Create form:

Edit Form:
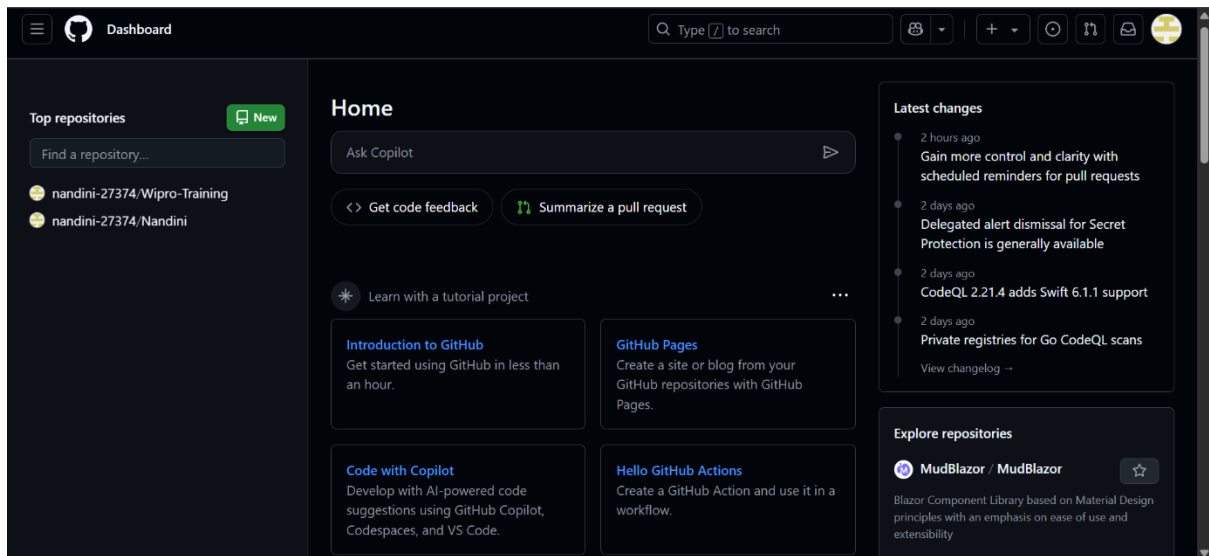

8.3 Deployment Steps:

Step-1 : Create a folder in the File Explorer.

Step-2 : Open that folder in Visual Studio Code.
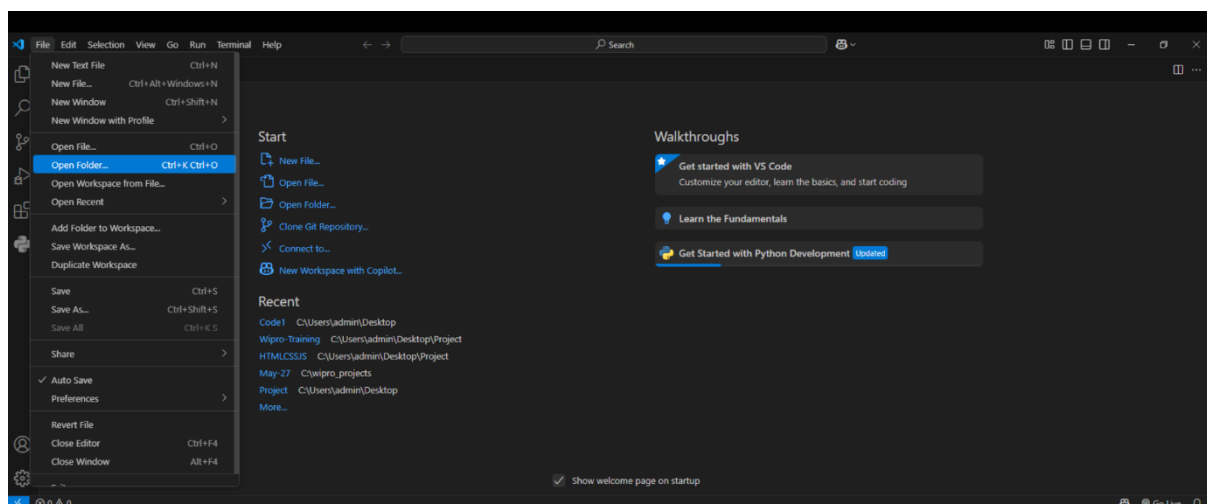
Step-3 : Go to the Git Hub home page

There we can see the "New" ,click on it it will navigate to another page.
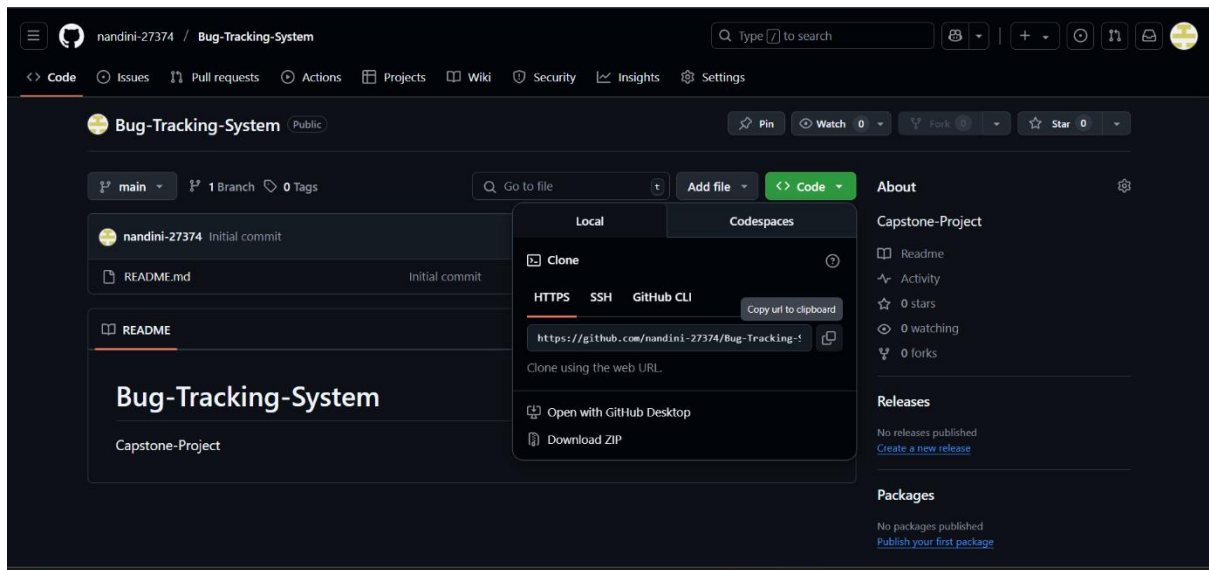

Step-4 : It will ask for Repository name after that, Description is optional.
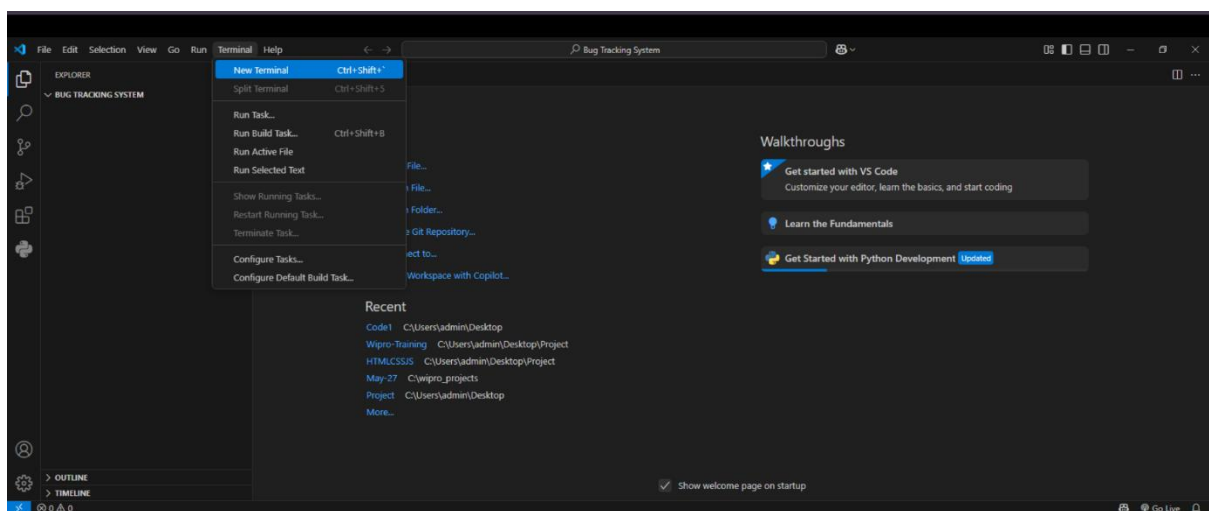
Here Repository name is "Bug Tracking System" and Description as "Capstone-Project".

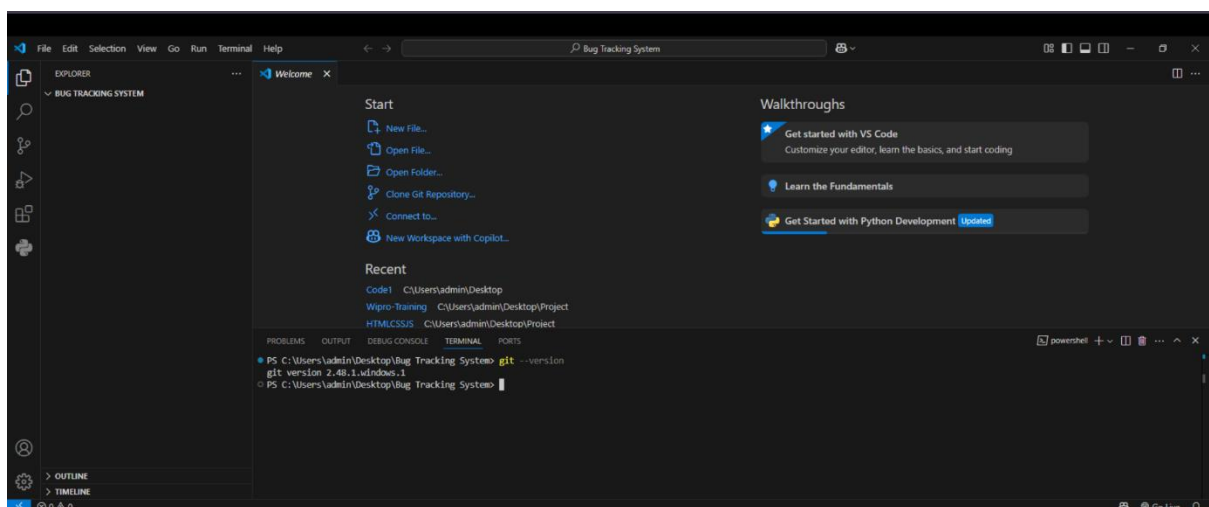Step-5 : Select the Public and README file, then Click on "Create Repository".

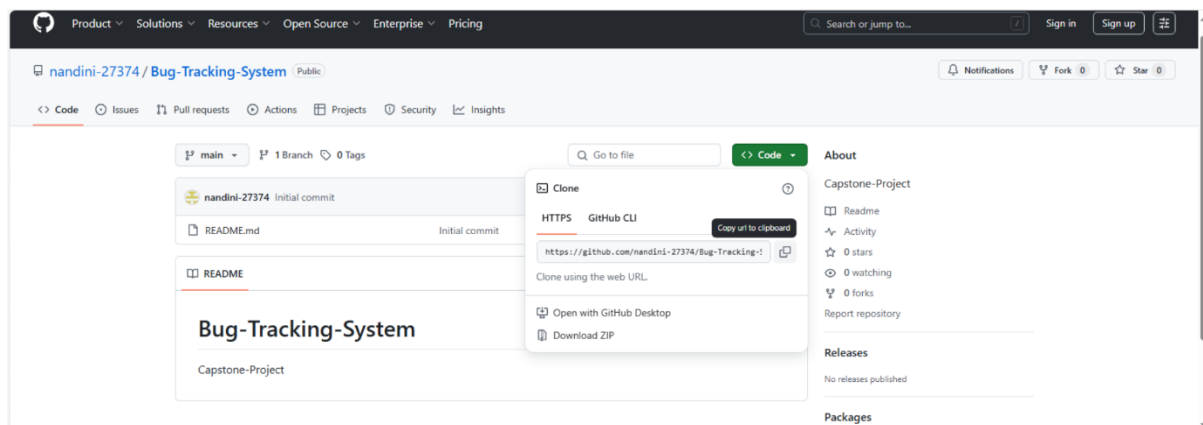A new repository is created with name Bug Tracking System.

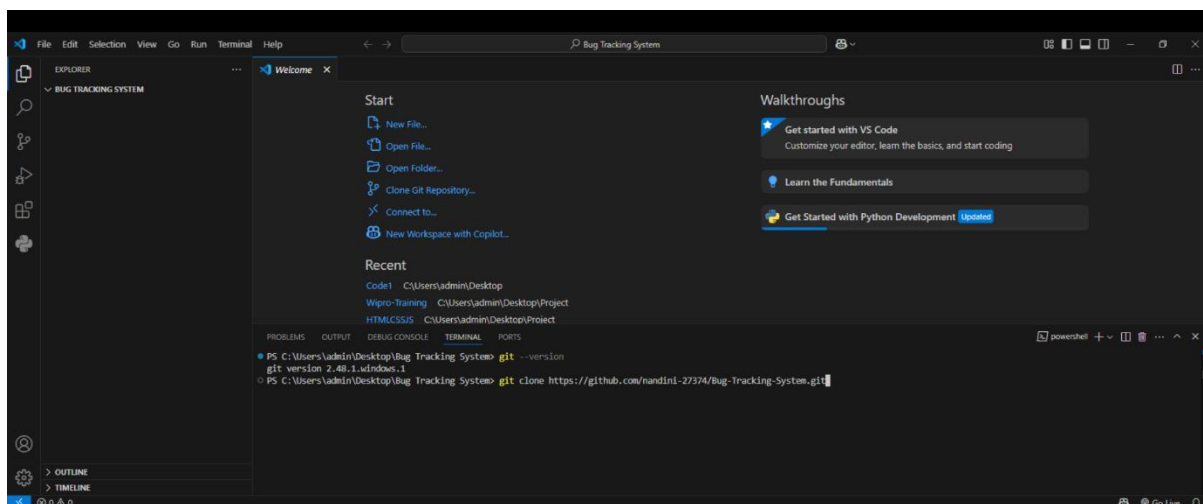Step-6 : Go to the VS Code then open the new terminal

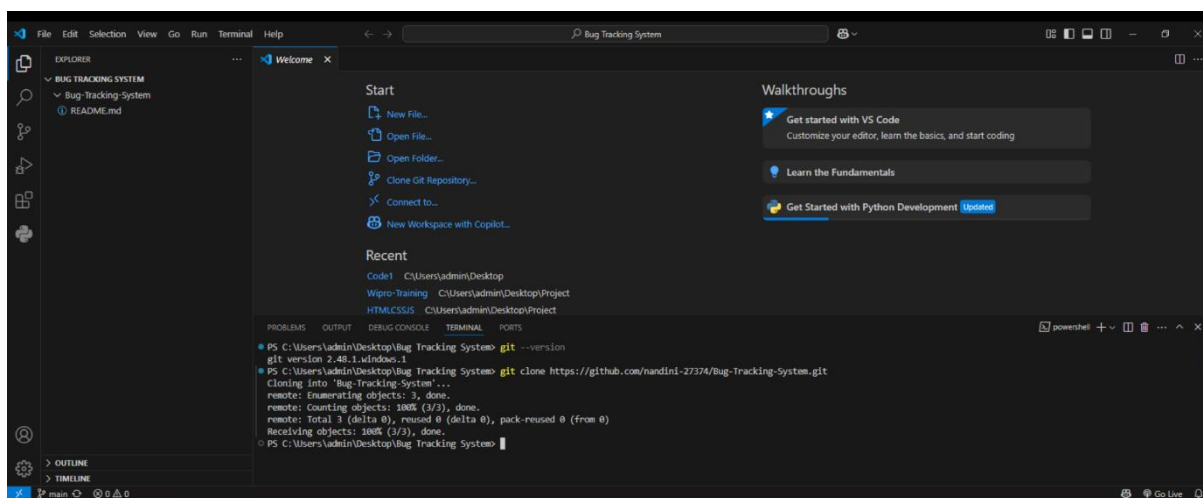After clicking new terminal a new window will come.

Step-7 : Go to the Git Hub newly created repository and copy that clone link.

Step-8 : Now come to Visual Studio code , In terminal paste the link.


Now give enter.

Now we can see at the left side corner we can see the "Bug Tracking System" which we have created in the Git Hub.

Step-9 : Copy the files that you want to push into the GIT Hub.

- Go to admin in the desktop there search for source from there you can find the PROJECTS web application.

- And also copy the word doc containing setup and usage instructions, Deployment steps, Screenshots of the outputs also ppt .
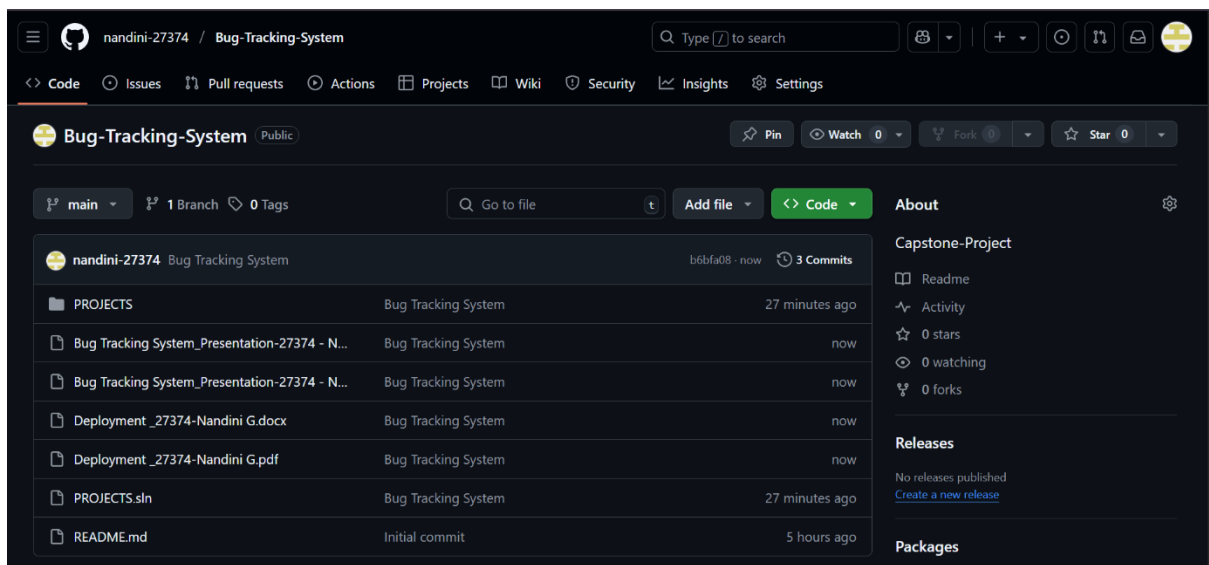
- Paste into Bug Tracking System folder.
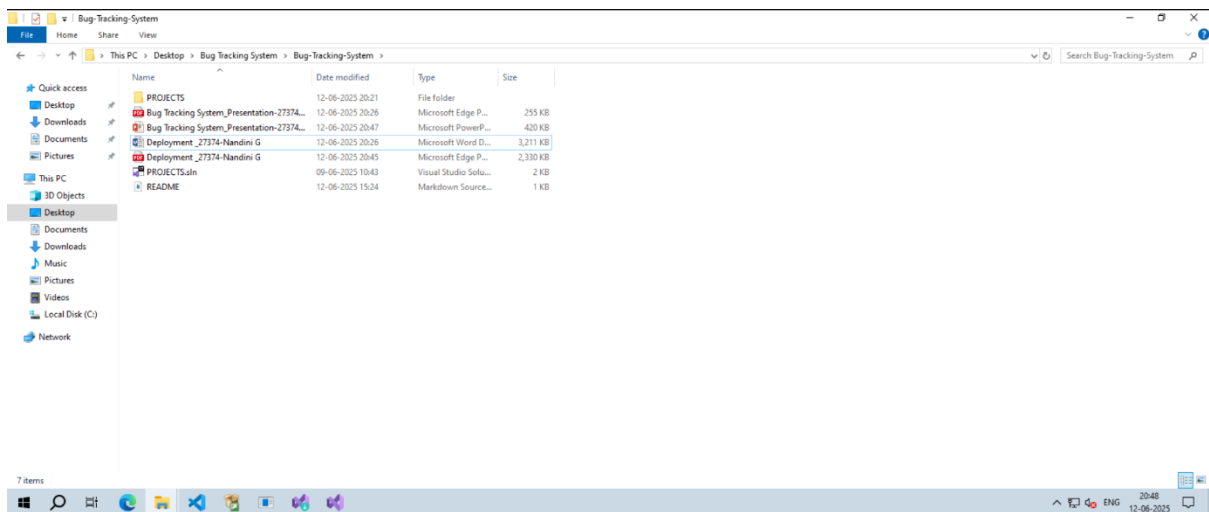
Step-10 : Go to visual studio code there you can see source control is having some action.
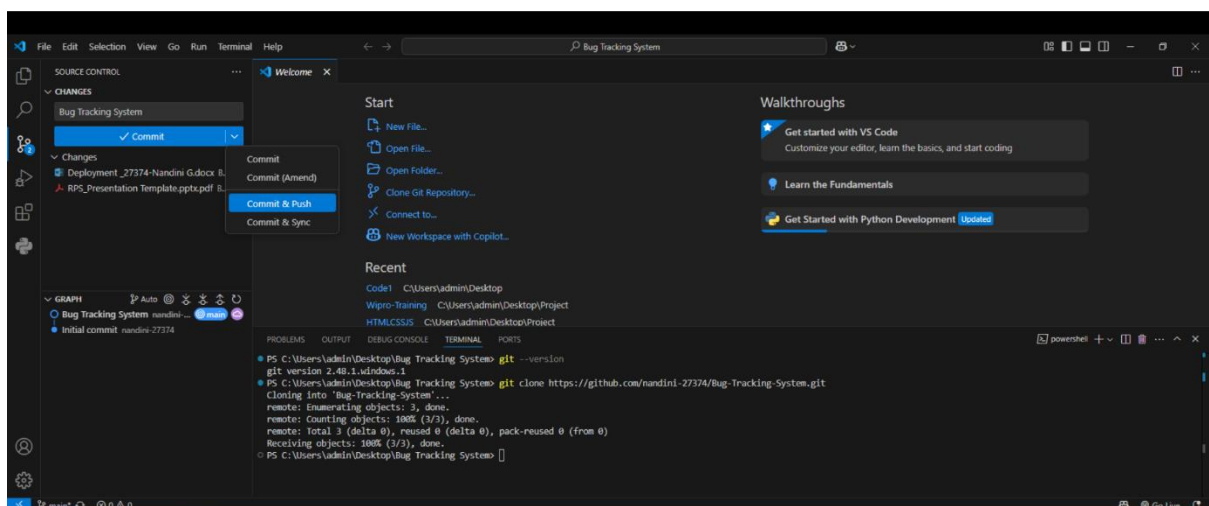
There give a commit and select commit & push.

Refresh the GIT Hub you can see the files there.

These are the files that we pushes through the Visual Studio Code.

By

G Nandini – 27374

.Net FullStack

25VID0820