WIPRO NGA Program – .Net FullStack

Capstone Project Presentation – 14-06-2025

Project Title Here - BUG TRACKING SYSTEM (MVC - Single Controller)

Presented by - G NANDINI - 27374

Capstone Project: 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 interfac.

Table of Contents

RPS learn..evolve

Capstone Project: Bug Tracking System (MVC - Single Controller)

2. Technologies Used

ASP.NET Core MVC
Entity Framework Core
SQL Server
Razor Views + Bootstrap
Azure App Service (Deployment)
Git for version control

Capstone Project: Bug Tracking System (MVC - Single Controller)

3. Database Schema:

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

Capstone Project: Bug Tracking System (MVC - Single Controller)

4. Controller

- BugsController.cs

Handles:

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

Capstone Project: Bug Tracking System (MVC - Single Controller)

5. Views Views/Bugs/

```
├── Index.cshtml
├── Create.cshtml
├── Edit.cshtml
├── Details.cshtml
└── Delete.cshtml
```

Capstone Project: Bug Tracking System (MVC - Single Controller)

6. Folder Structure

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

Capstone Project: Bug Tracking System (MVC - Single Controller)

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

Introduction

Bug tracking systems are a crucial part of any software development team. Without a good bug tracking systems, you may find it difficult to track the progress of your projects, identify and fix bugs quickly, and provide a quality product to your customers.

What is Bug Tracking?

Bug tracking is the process of documenting and managing the issues that occur while using a software application. Bugs can be categorized as functional, nonfunctional, or unknown.

Functional bugs refer to problems with the user interface or functionality of the application. Nonfunctional bugs refer to problems with how the application behaves, such as errors or crashes. Unknown bugs are those that have not been assigned a particular bug category.

Introduction

What is Bug Tracking System ?

A Bug Tracking System is a software application that helps you track issues and bugs in your software. It can help you to resolve issues quickly and efficiently, and it can also help you to improve the quality of your software.

There are a few different types of bug tracking systems, but the most reliable one is probably Disbug. Disbug offers a chrome extension and a website feedback widget which smoothes the entire process of bug reporting, and it is used by hundreds of software teams all over the world.

Bug tracking systems are essential for any software company, and they can help you to improve your product quality, customer support, and overall profitability.

Common types of Bugs

- Syntax bugs: Breaking the rules of the programming language.
- Logical bugs: Code runs without errors but produces incorrect results.
- Runtime errors: Crashes during execution (e.g., null references, division by zero).
- UI bugs: Elements not displaying or behaving correctly.
- Integration bugs: Issues when different systems or modules interact.
- Performance Bugs: Slow execution, excessive memory or CPU usage, lagging user interactions—often due to inefficient algorithms, memory leaks, or poor caching/practices
- Compatibility Bugs: Software works in one environment (browser, OS, device) but fails or behaves unexpectedly in others—think CSS issues across browsers or OS-specific behaviors .

Benefits of using a bug tracking system

Bug tracking systems have many benefits that can make your life as a software developer or software tester much easier. Here are just a few of the most important ones:

1. Bug tracking systems help you stay organized. Since you can track all the bugs that are related to a specific project, you will be able to identify and fix them faster. This will save you time and keep your codebase clean and tidy.

2. Bug tracking systems help you track the progress of projects. By knowing which bugs have been fixed and which ones are still pending, you will be able to better measure the success or failure of a project. This is especially important if you are working on a project that has deadlines attached to it.

Benefits of using a bug tracking system

3. Bug tracking systems help you findbugs quickly. Since all the bugs related to a particular project are stored in one place, it will be easy to find the bug that you're looking for. This can save you a lot of time and hassle when trying to track down a bug that is causing problems in your codebase.

4. Bug tracking systems provide feedback to developers and testers. By automatically sending notifications whenever a bug is fixed or new bugs are found, bug tracking systems help developers and testers receive feedback as soon as possible. This helps them to better understand the issues that they're facing and resolves them faster - saving both time and money in the process.

What should a bug tracking system include?

A bug tracking system is a great way to keep track of all the bugs that are reported by your users. It ensures that the issues that are reported are properly documented, tracked, and resolved as quickly as possible. Here are a few features that a good bug tracking system should have:

1. A user-friendly interface - The interface should be easy to use and intuitive, so that everyone in the company can access it.

2. Automatic email notification - Whenever a bug is reported or resolved, the system should send an email notification to all involved parties. This will help keep everyone informed and focused on the issues at hand.

What should a bug tracking system include?

3. Detailed reports - The reports should include all the information necessary to understand the bug history, such as the date, time, user information, and more. This will help you track trends and make informed decisions about how to handle future issues.

4. Automated resolution workflow - The system should have an automated resolution workflow so that all bugs are resolved as quickly as possible. This will help to prevent any major delays or disruptions in your workflow and ensure that your users are always happy and satisfied with your product.

RPS learn..evolve

Steps to follow for doing the Project:

Step-1: Open the Visual Studio 2022 and click on Create New Project.

Create the new project names with PROJECTS.

Step-2 : In model create a class with Bug.cs and define the fields in that.

Step-3 : Create a controller.

Step-4 : In package manager console run the below commands.

Add-Migration InitialCreate

Update-Database

Step-5 :Run the controller a basic web application will run in the browser.

Steps to follow for doing the Project:

Step-6 : Add the style part to the code for

Views/Bugs/Index.cshtml

Views/Bugs/Edit.cshtml

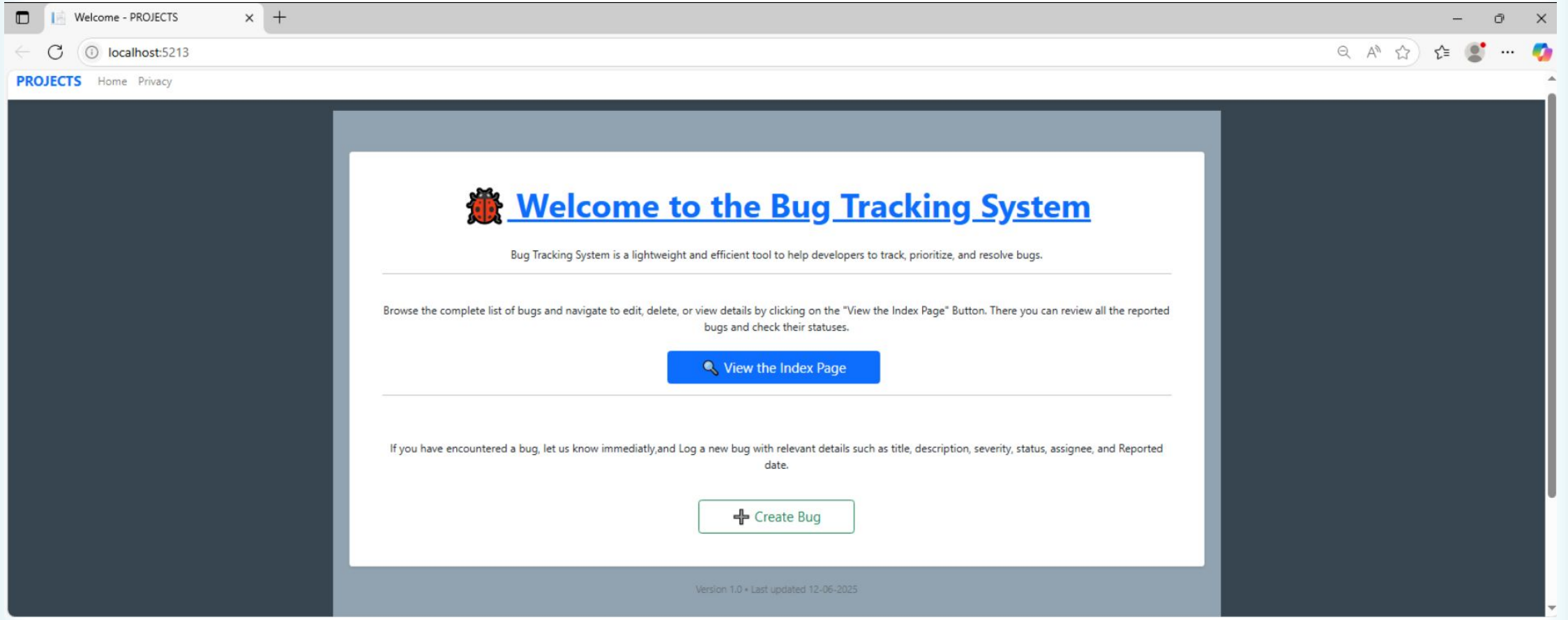
Views/Bugs/Details.cshtml

Views/Bugs/Create.cshtml

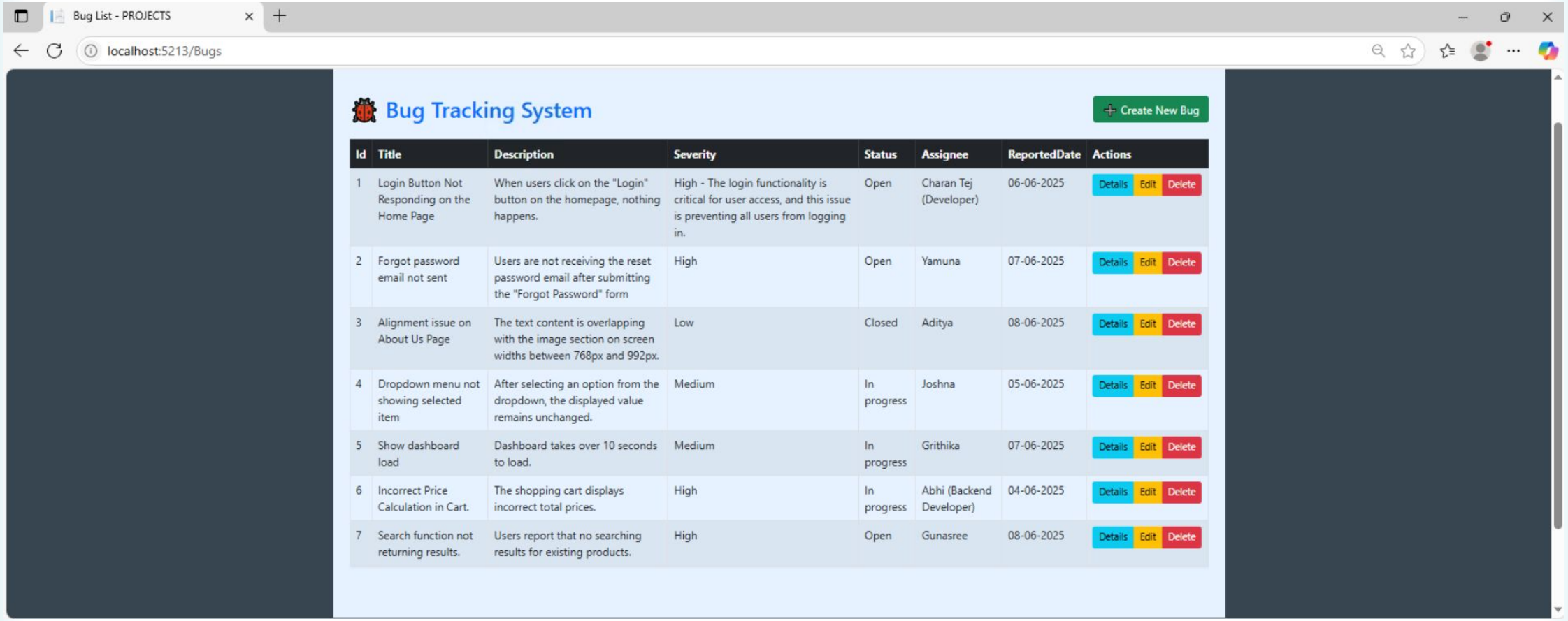
Views/Bugs/Delete.cshtml

Step-7 : Modify the code for Views/Home/Index.cshtml, then run the code.

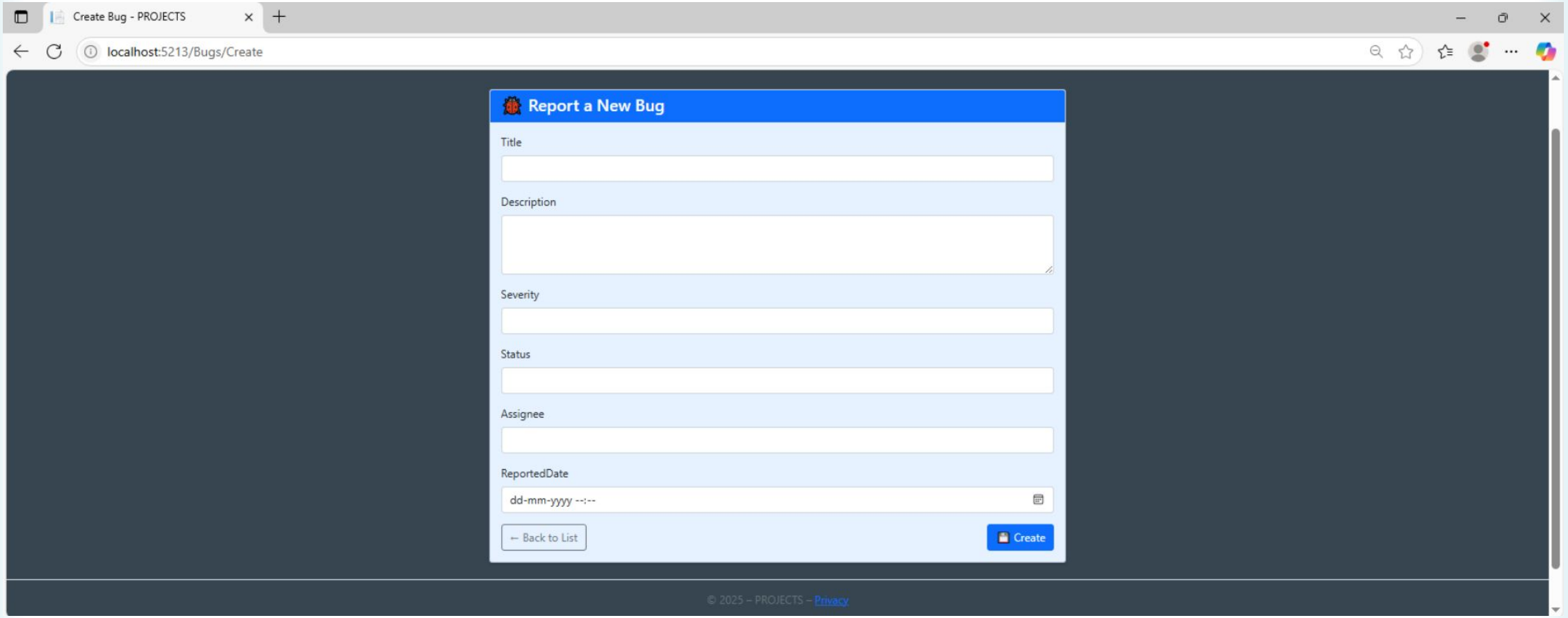
Home page - Output



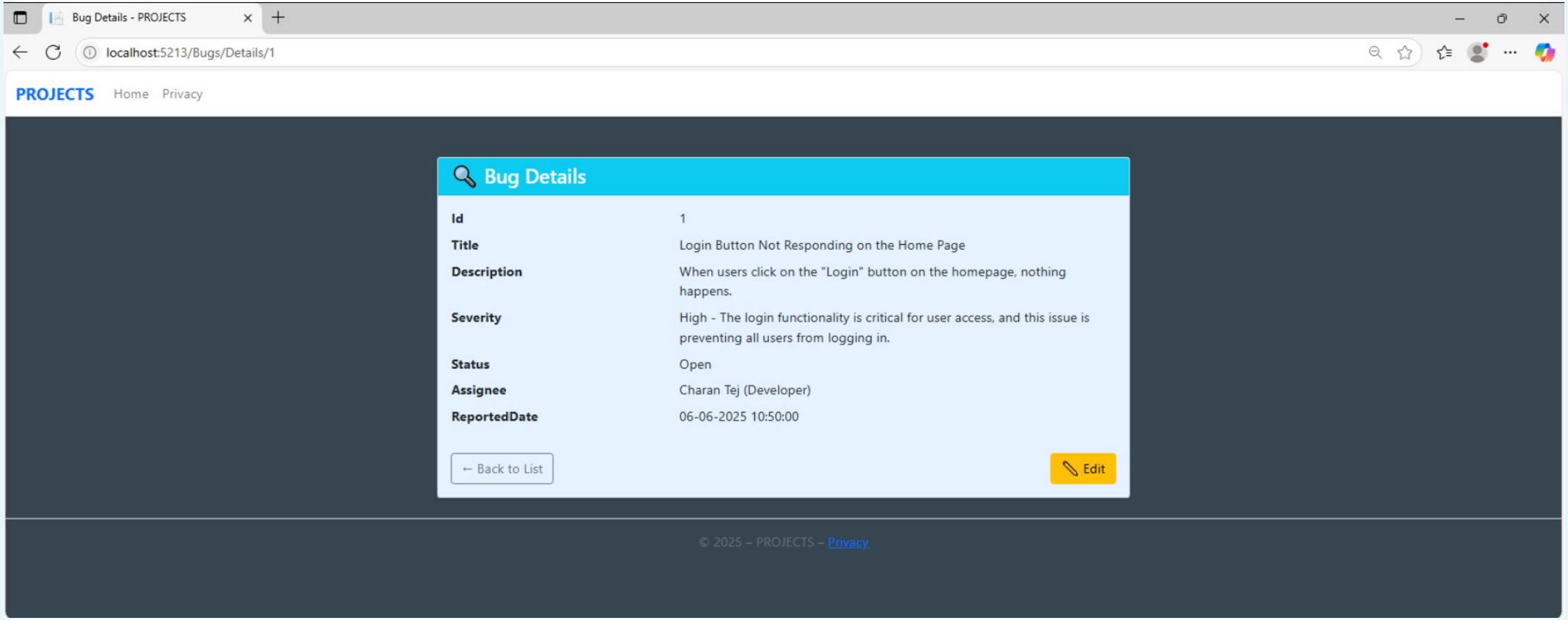
Index Page



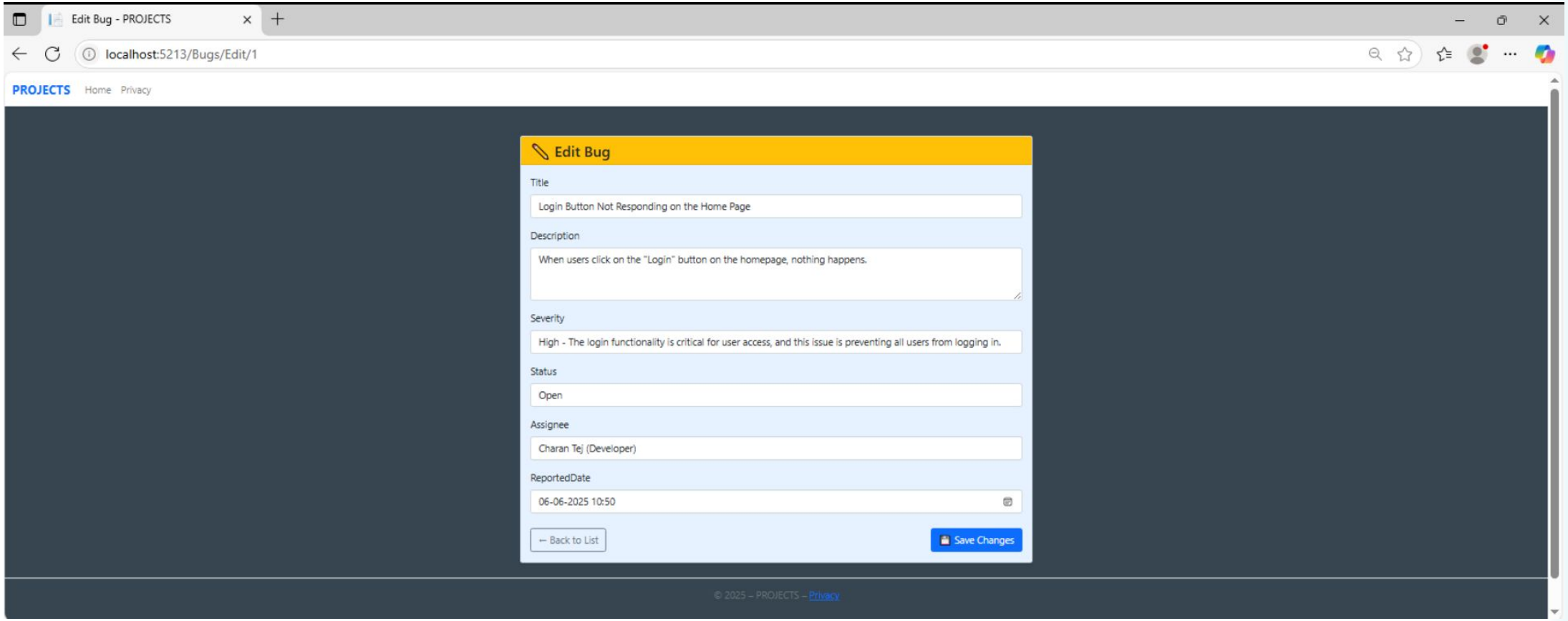
Create page



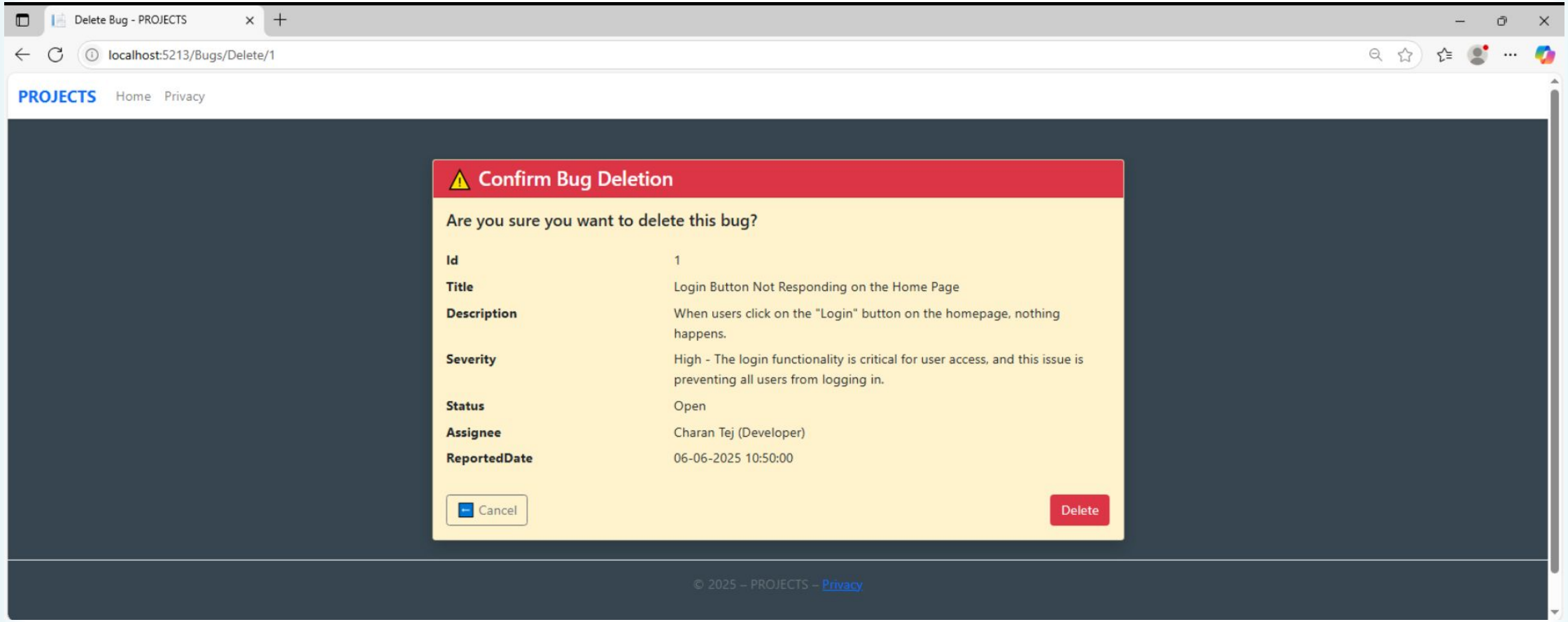
Details page



Edit page



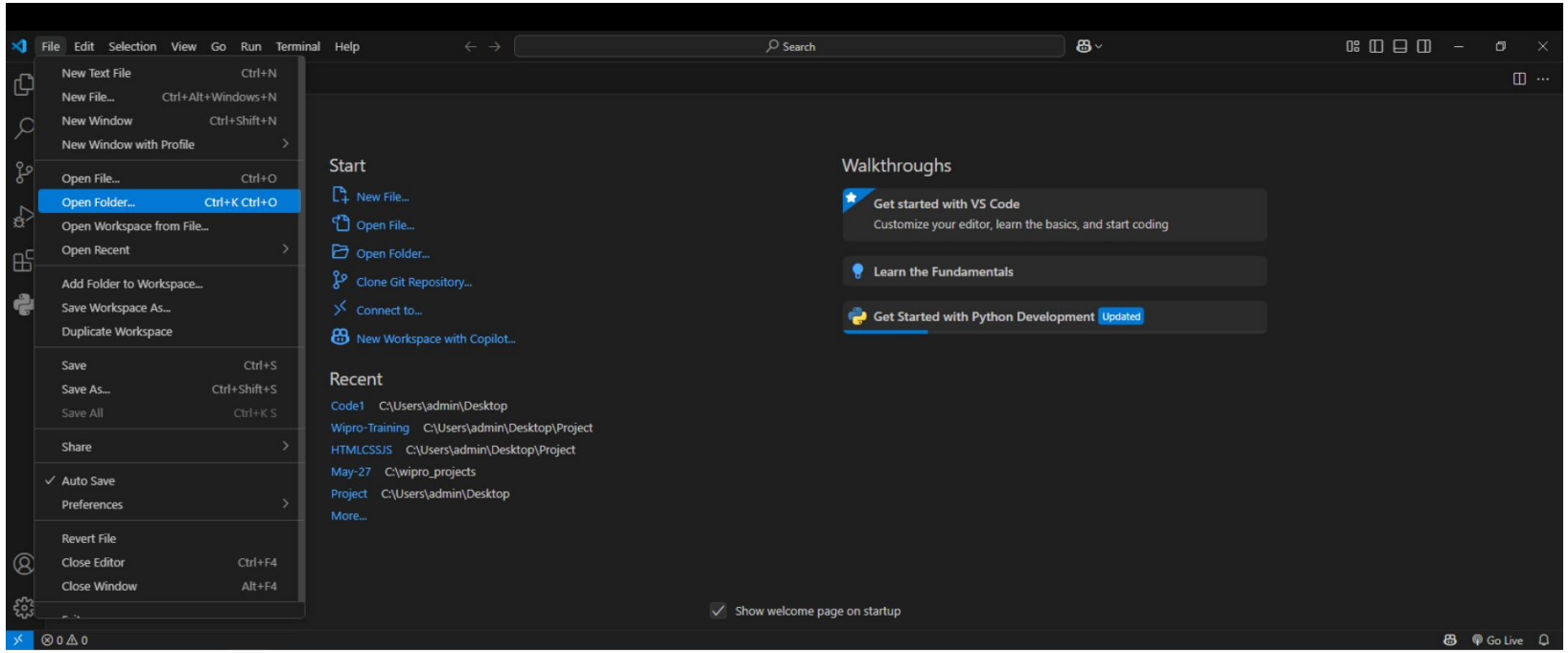
Delete page



Deployment Steps

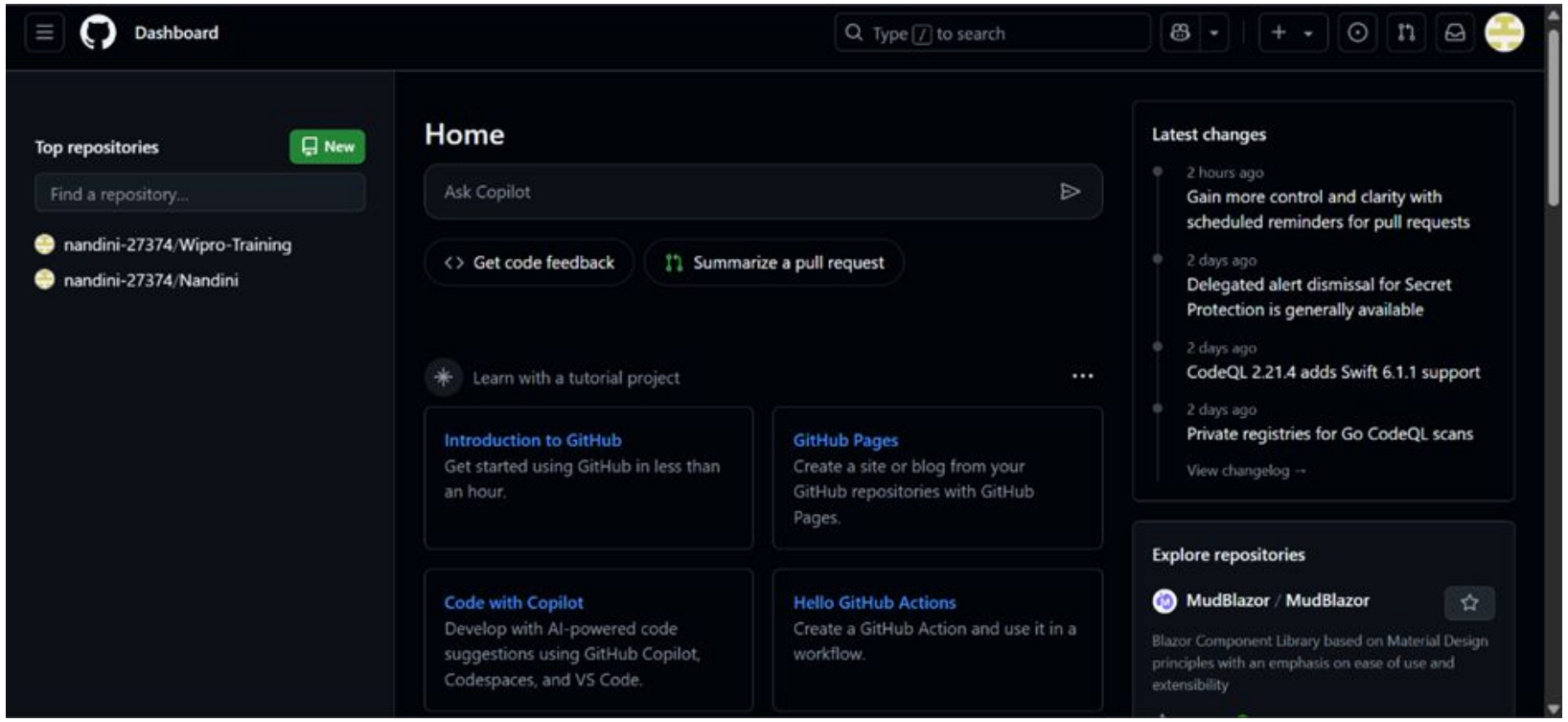
Step-1 : Create a folder in the File Explorer.

Step-2 : Open that folder in Visual Studio Code.



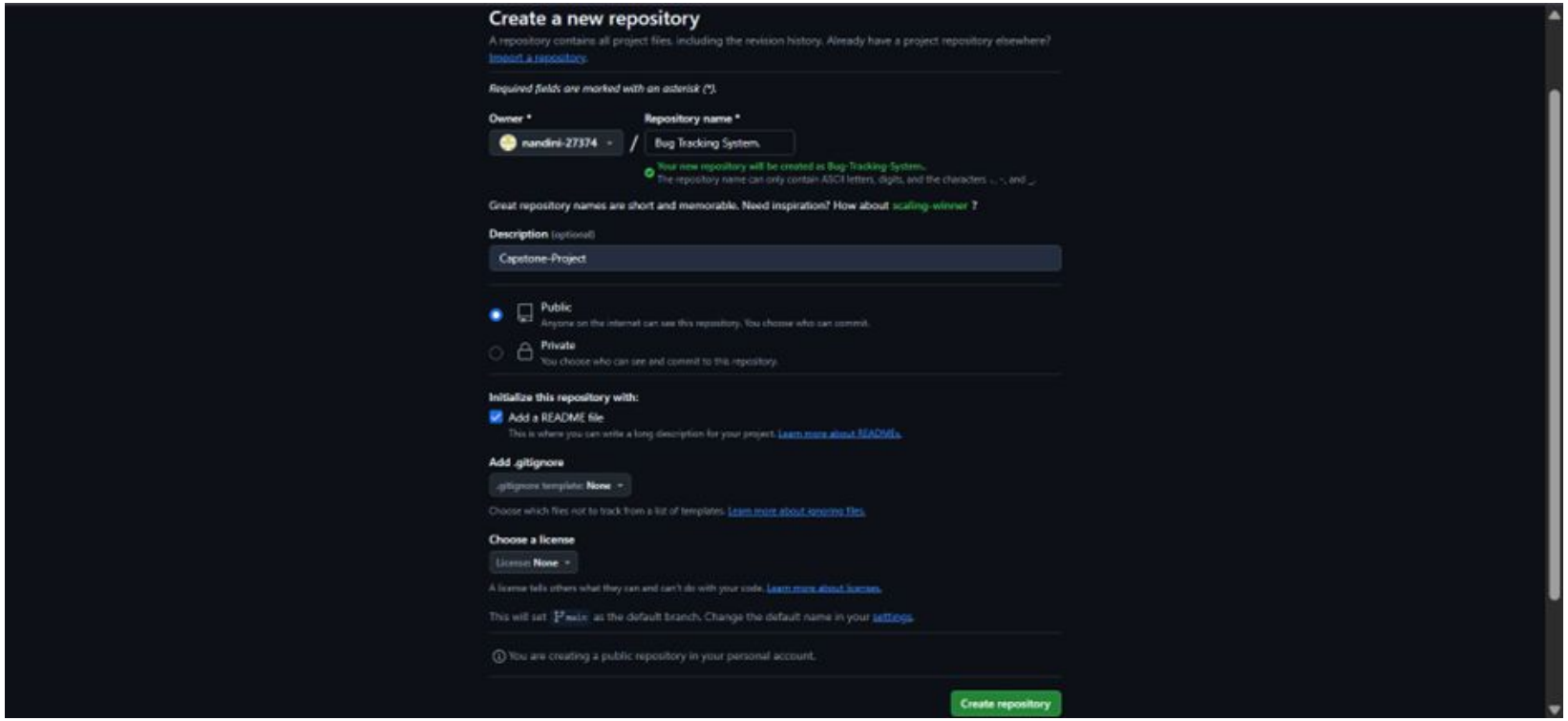
Deployment Steps

Step-3 : Go to the Git Hub home page



Deployment Steps

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



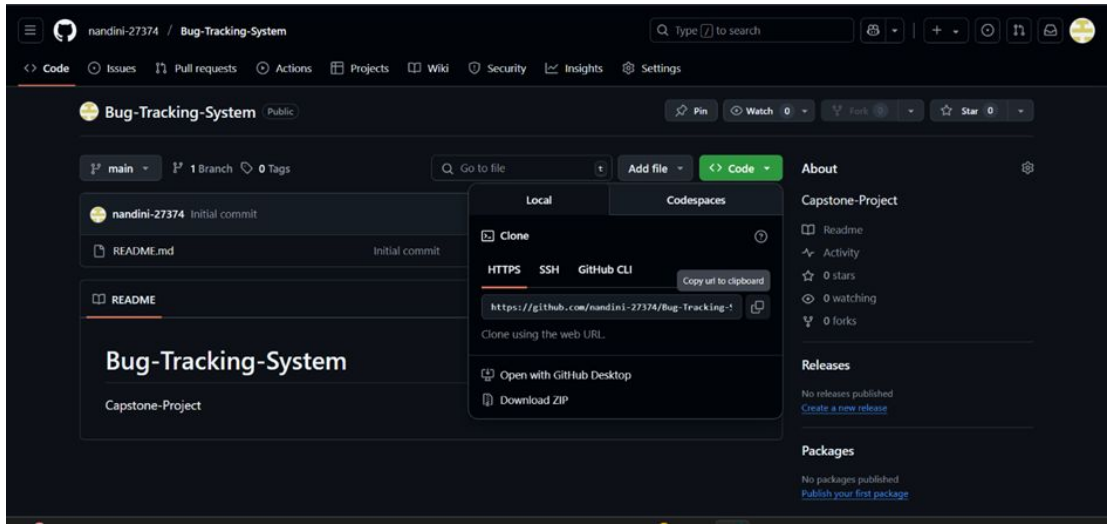
Deployment Steps

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”.

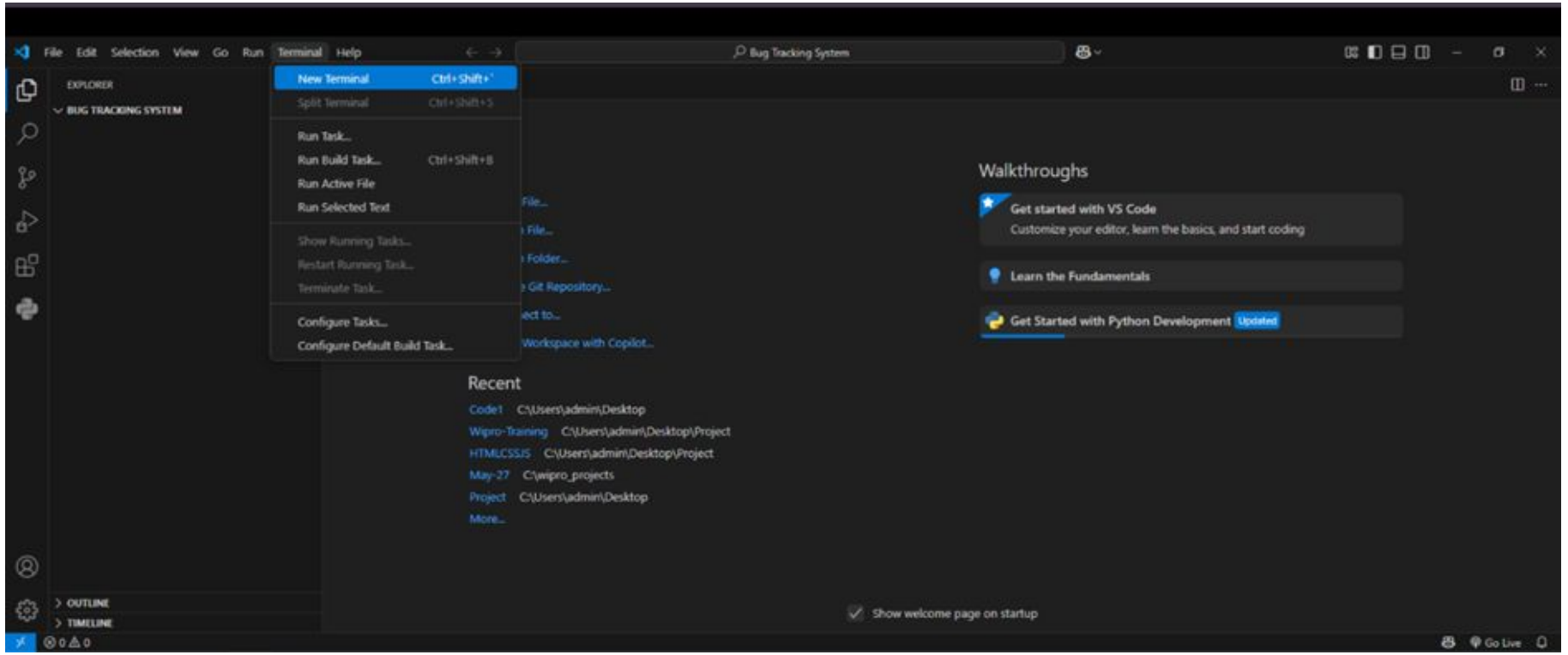
Step-5 : Select the Public and README file, then Click on “Create Repository”.

A new repository is created with name Bug Tracking System.



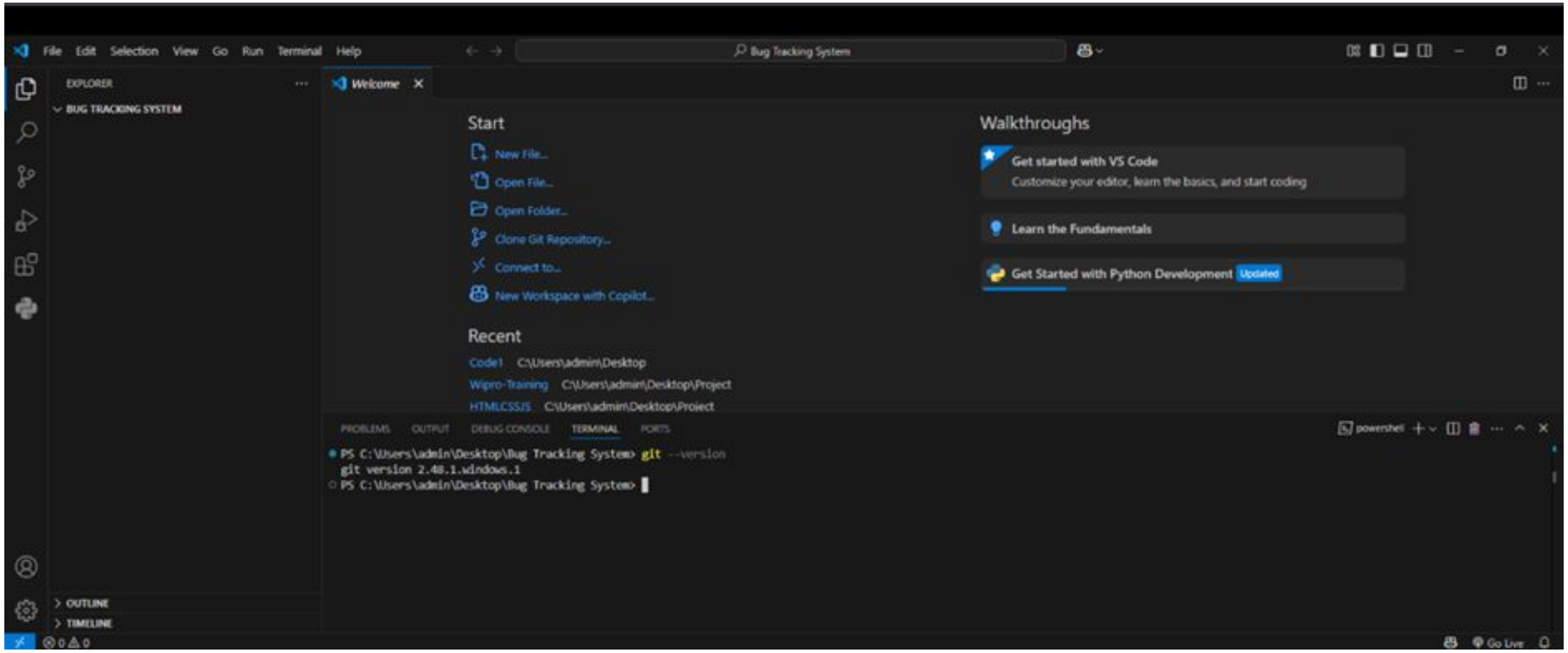
Deployment Steps

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



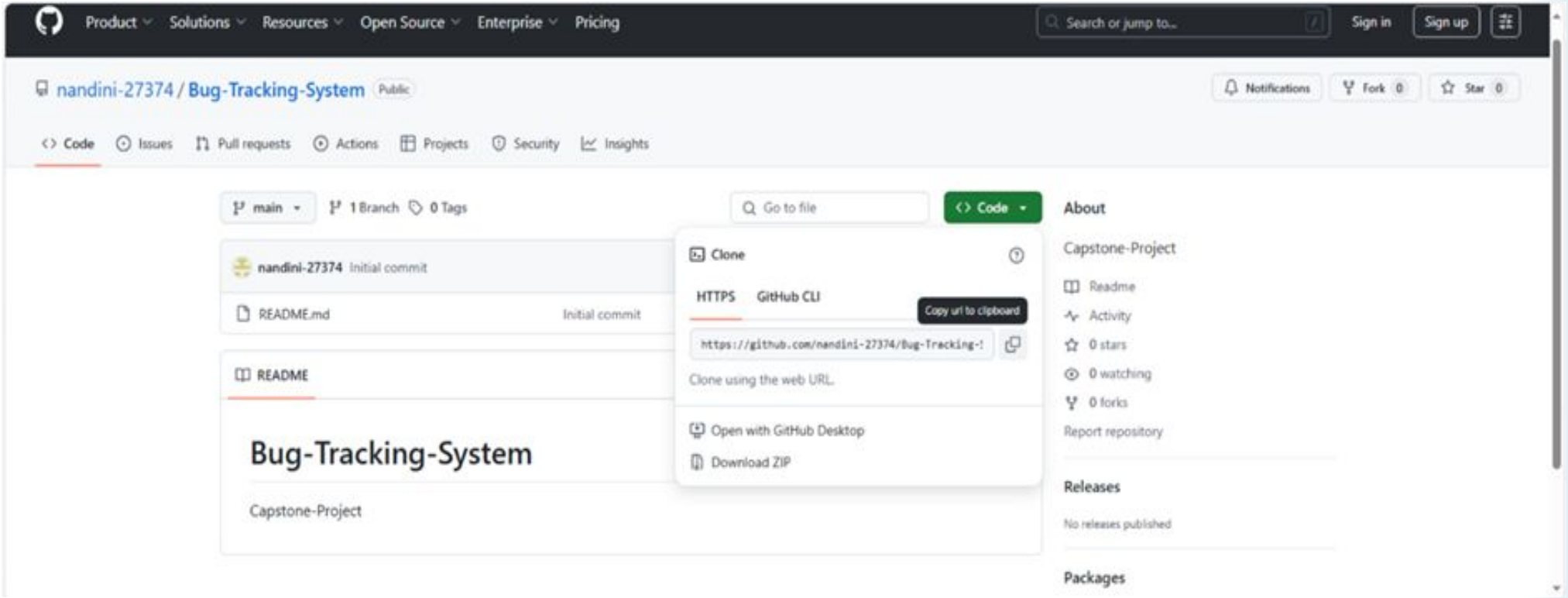
Deployment Steps

After clicking new terminal a new window will come.



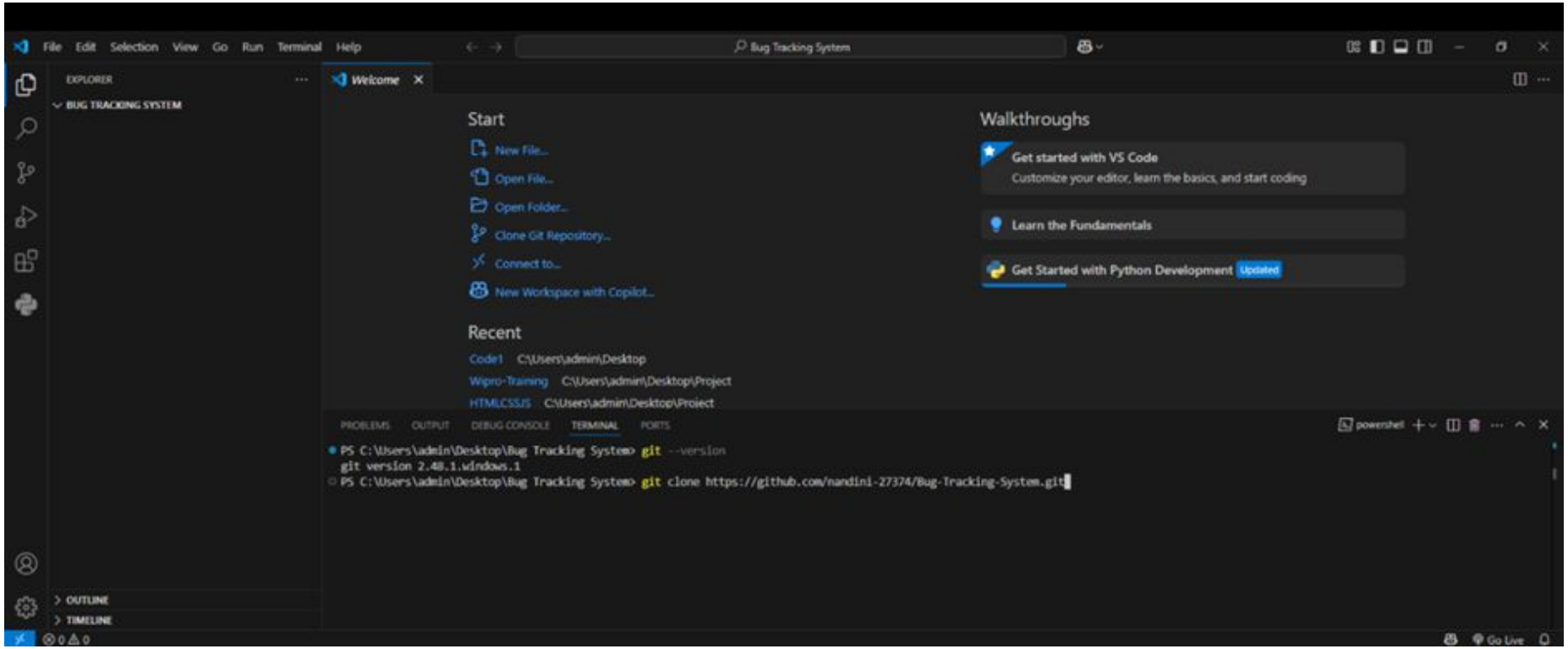
Deployment Steps

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



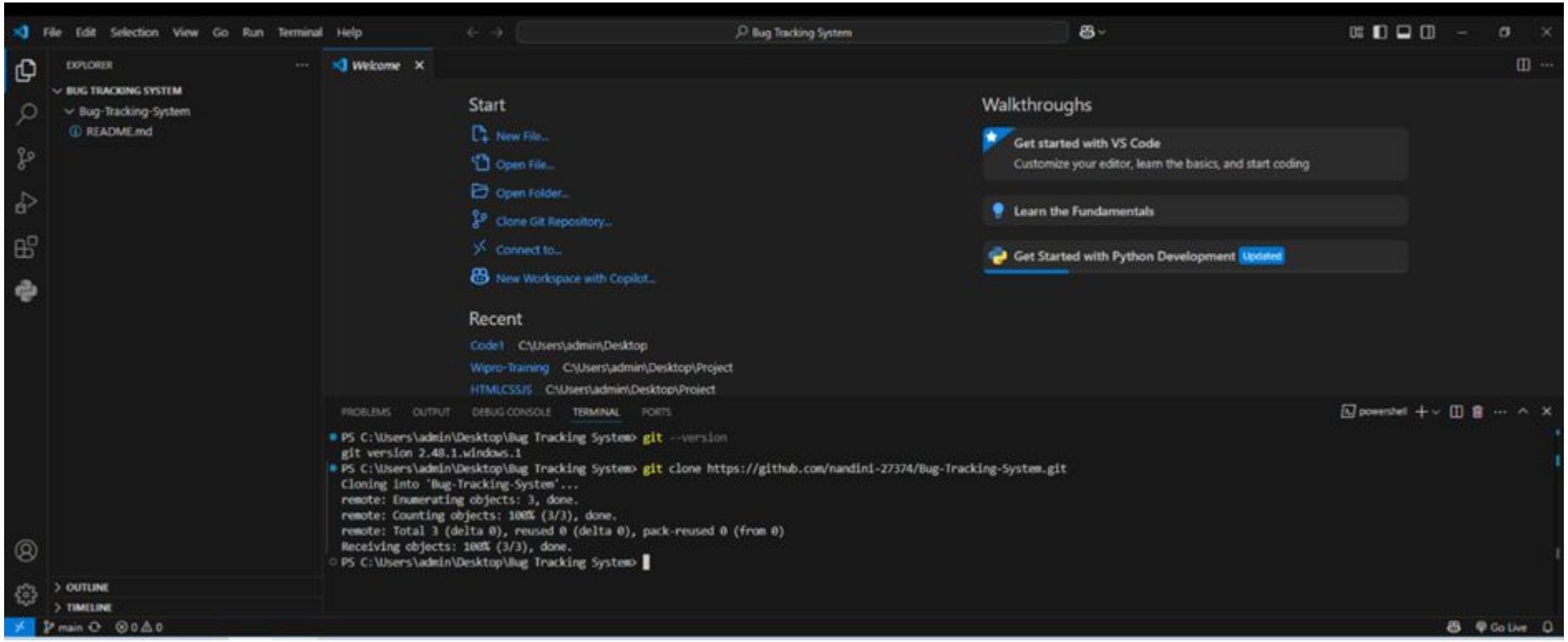
Deployment Steps

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



Deployment Steps

Now give enter.



learn..evolve

Deployment Steps

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

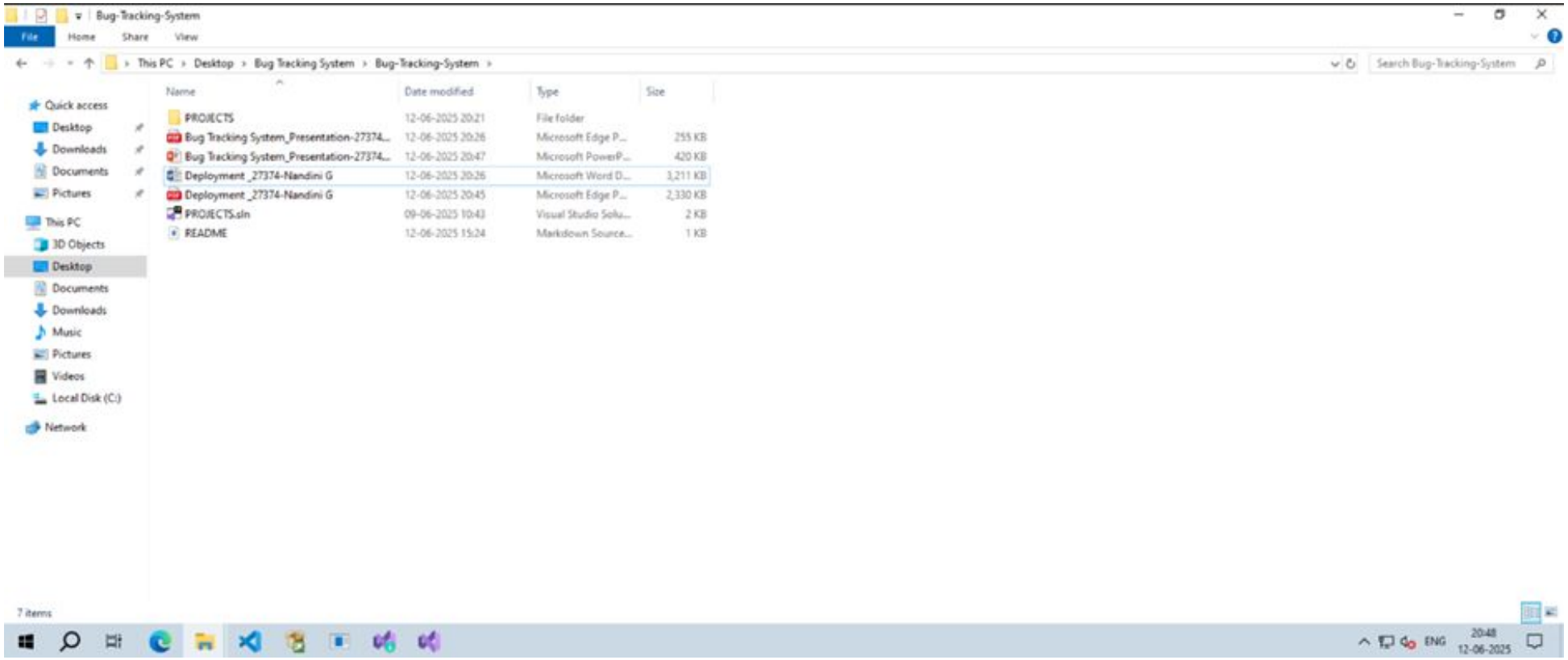
PROJECTS web application.

And also copy the word doc containing setup and usage instructions, Deployment steps,

Screenshots of the outputs also ppt .

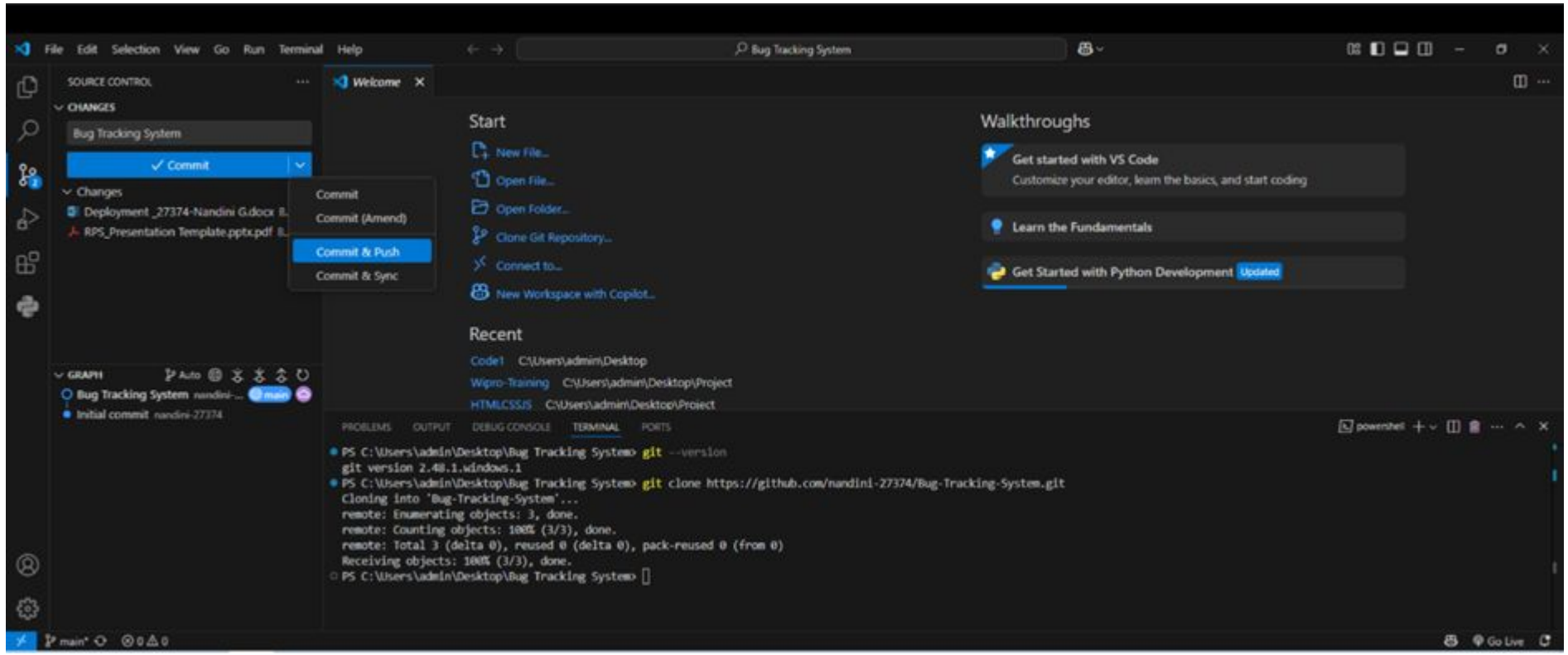
Paste into Bug Tracking System folder.

Deployment Steps



Deployment Steps

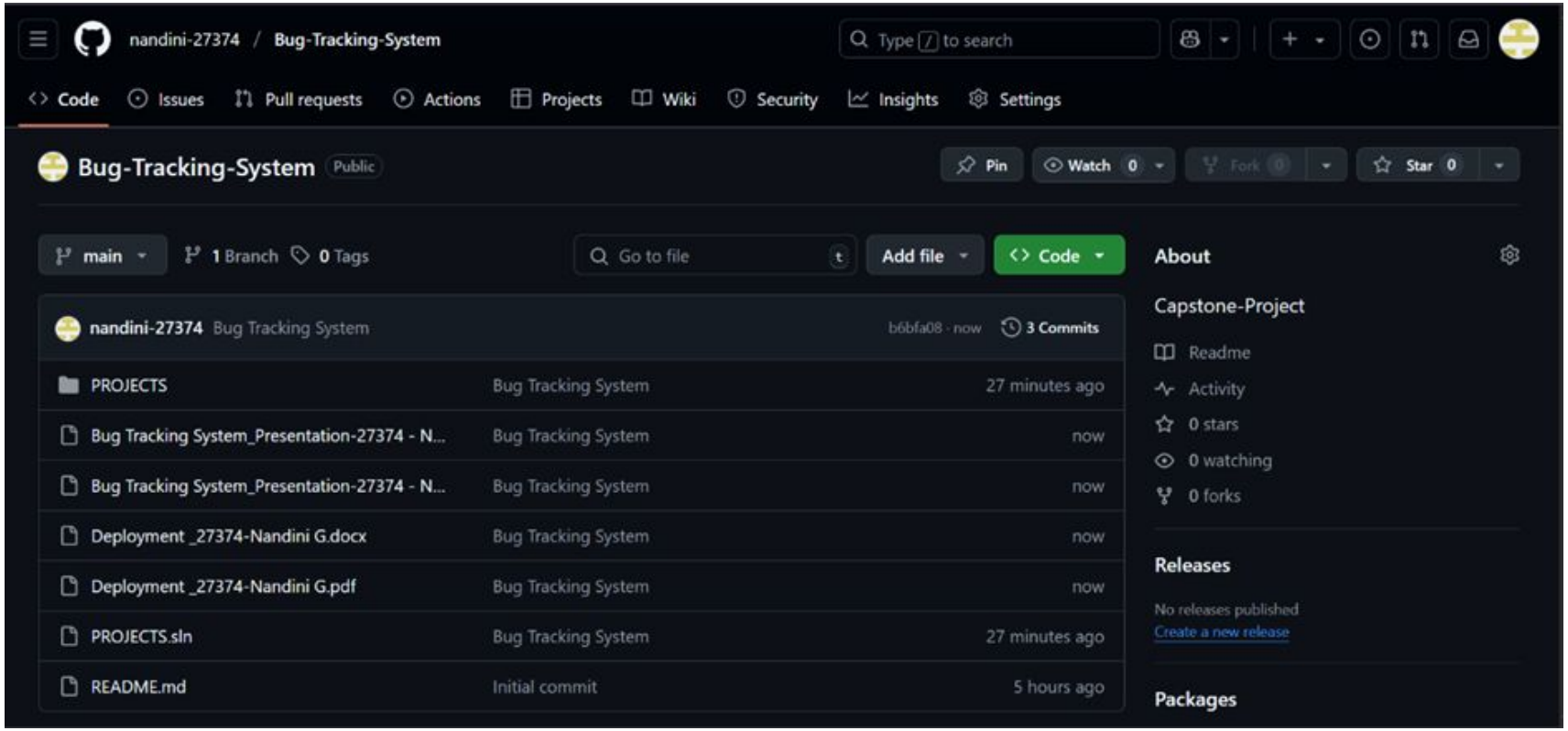
Step-10 : Go to visual studio code there you can see source control is having some action.



Deployment Steps

There give a commit and select commit & push.

Refresh the GIT Hub you can see the files there.



Conclusion

The whole process of tracking bugs is not easy, especially when it comes to managing hundreds of issues at the same time. Hence we have listed down everything that you need to know about bug-tracking systems as well as how and why they can help.

These solutions can keep your team focused on quality work and ensure timely resolution of all the issues you face. Thus, make sure not to skip a step in your bug-tracking journey. Keep reading our blog for more knowledge related articles.