**Team Members:**

**P. Mithravinda**

**E. Swathi**

**L. Indu**

**Y. Divya**

**G. Reshmaa Sree**

**S. Vanitha**

**DevOps**

DevOps is a technology that tells us how to approach process statement.

DevOps = Development + Operations

It is a culture + set if practices + tools that help: - Developers and Operations.

Work together, instead of working in silos.

**Benefits: -**

1. Delivers the features faster.

2. Fix bugs quickly.

3. Automate repetitive tasks.

DevOps Life Cycle:

It has 8 phases......!

|  |  |
| --- | --- |
| **Phase** | **Description** |
| Develop | Define what to build |
| Plan | Write the code |
| Build | Compile the code into deployable format (Machine Language) |
| Test | Check for bugs automatically |
| Release | Approve the build for deployment (Permissions from Quality Analyst) |
| Deploy | Push the app to servers or app store |
| Operate | Keep the app running, fix any bugs |
| Monitor | Analyse the performance and feedback |

Cycle continues: Feedback from users --> back to plan

**GitHub basics:**

**Git**: Version control system which is used to track the files and managing projects with in your local system.

**GitHub**: It is a container which is used to store projects online by using repositories.

**Real-time example:**

If you are working with a team on online shopping website:

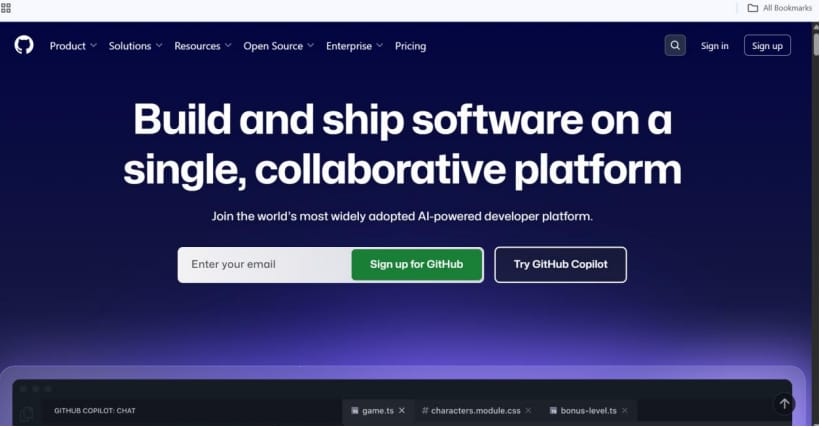
> You add "Wishlist" page.

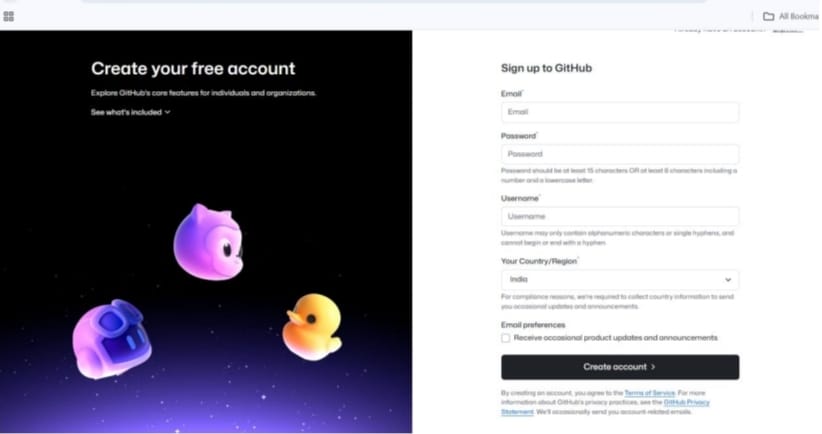
> GitHub helps us to merge, track and resolve conflicts if both change same code.

**Experiment 1:**

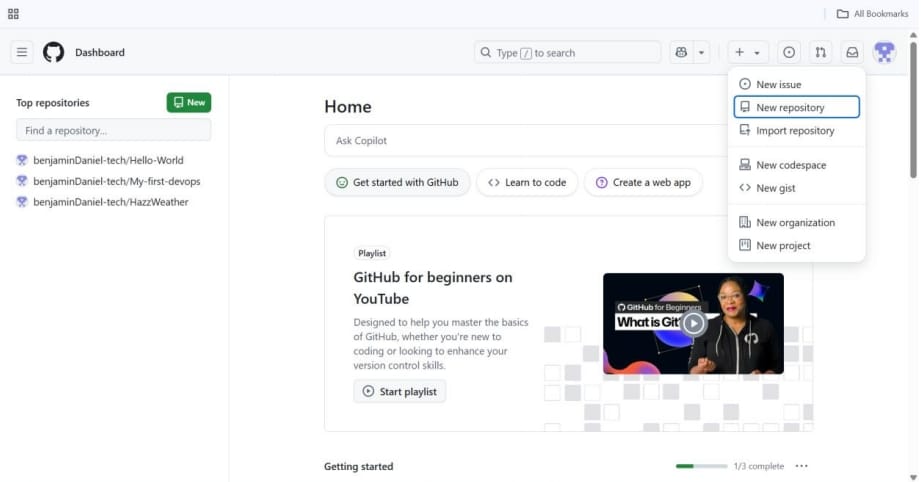
Create a GitHub account, then create a new repository where README file and upload a basic file named hello.txt, edit the file through GitHub’s web interface, commit the changes and view the commit history. Share the link to your repository after completion.

Step 1: Create GitHub account.

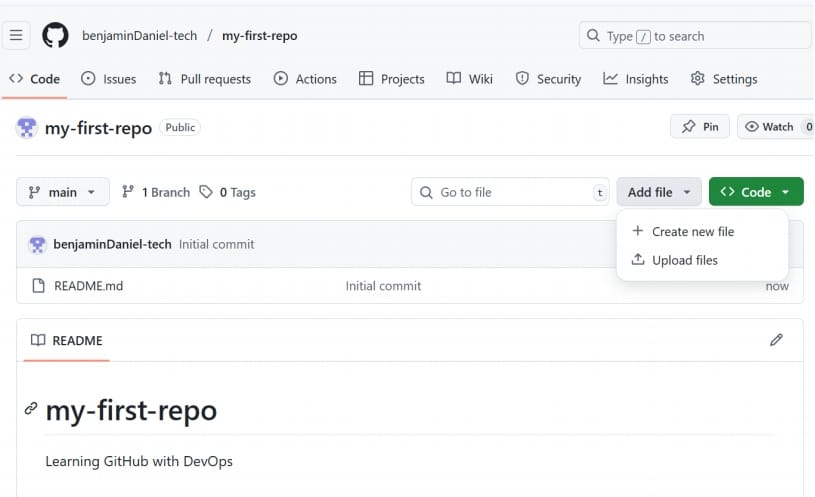




Step 2: Create a repository.



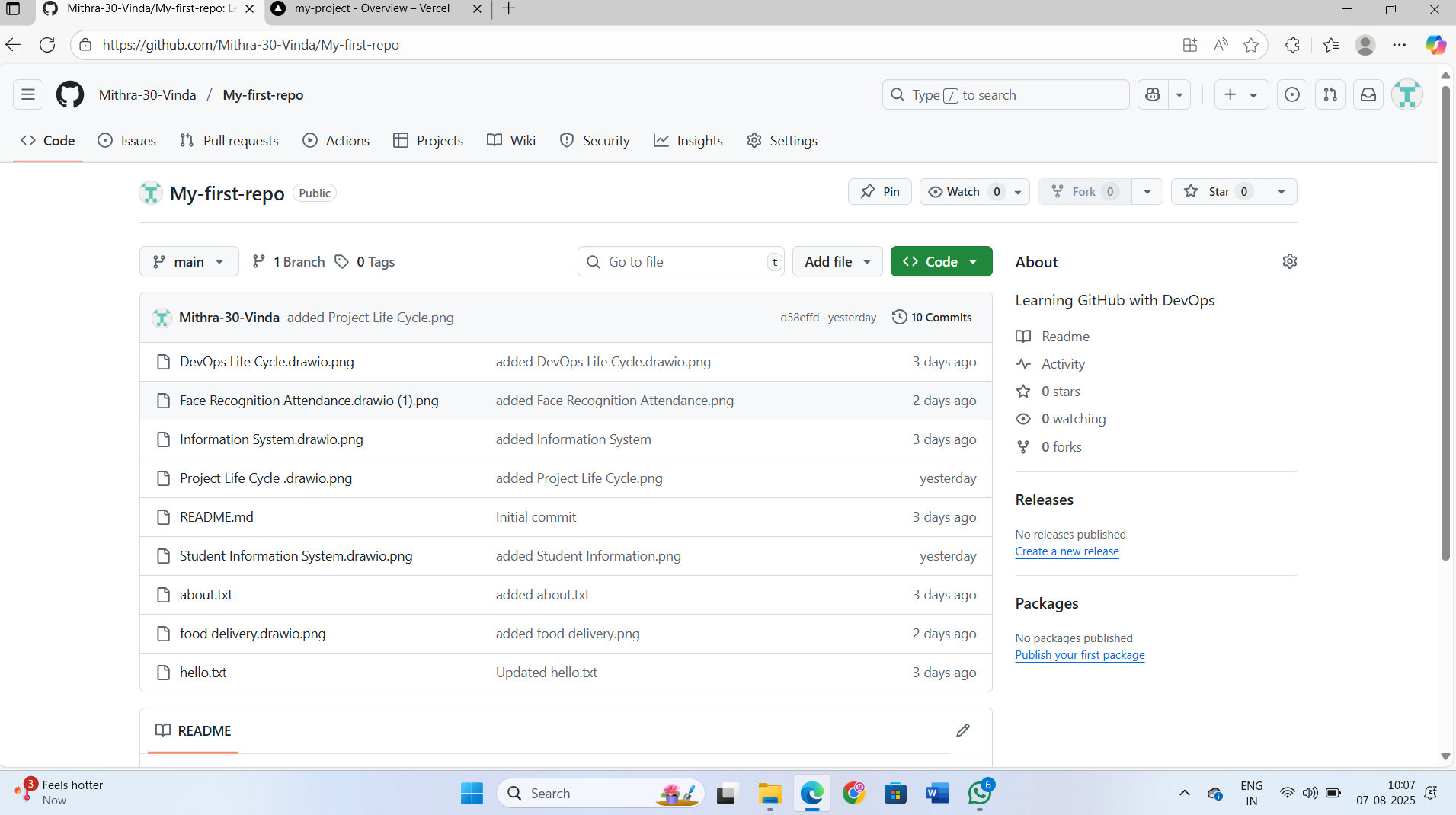
Step 3: Upload a file.



Step 4: Edit the file online.

Step 5: View version history.

Step 6: Submission.



> Copy and share your repo link....,

Example:

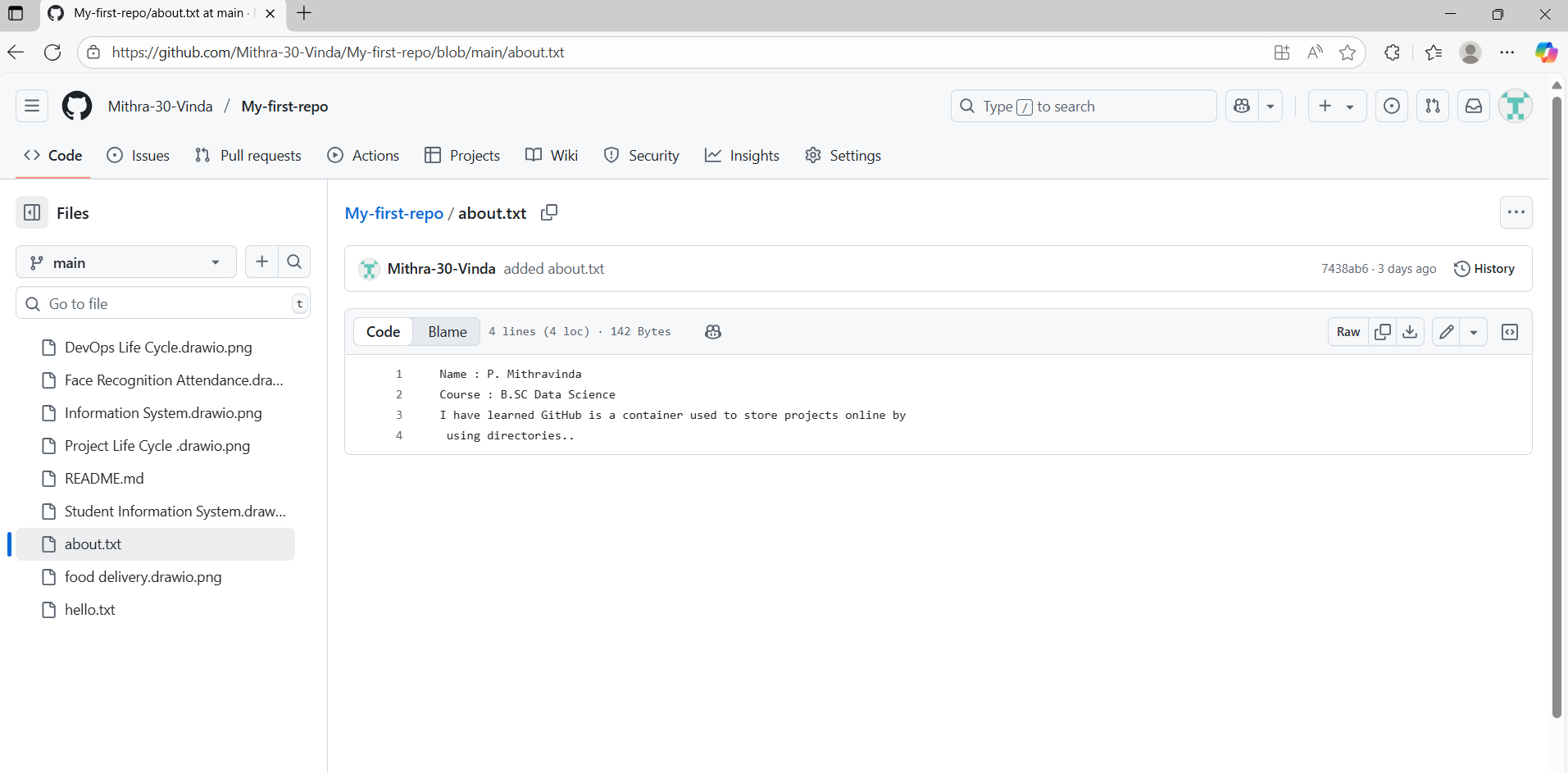
<https://github.com/Mithra-30-Vinda/My-first-repo>

**Experiment 2:**

Create a new file named as about.txt in your GitHub repository.

Inside the file, write your name, course, one thing you learnt about GitHub.

Commit the file to your repository and share the link after completion of the task.

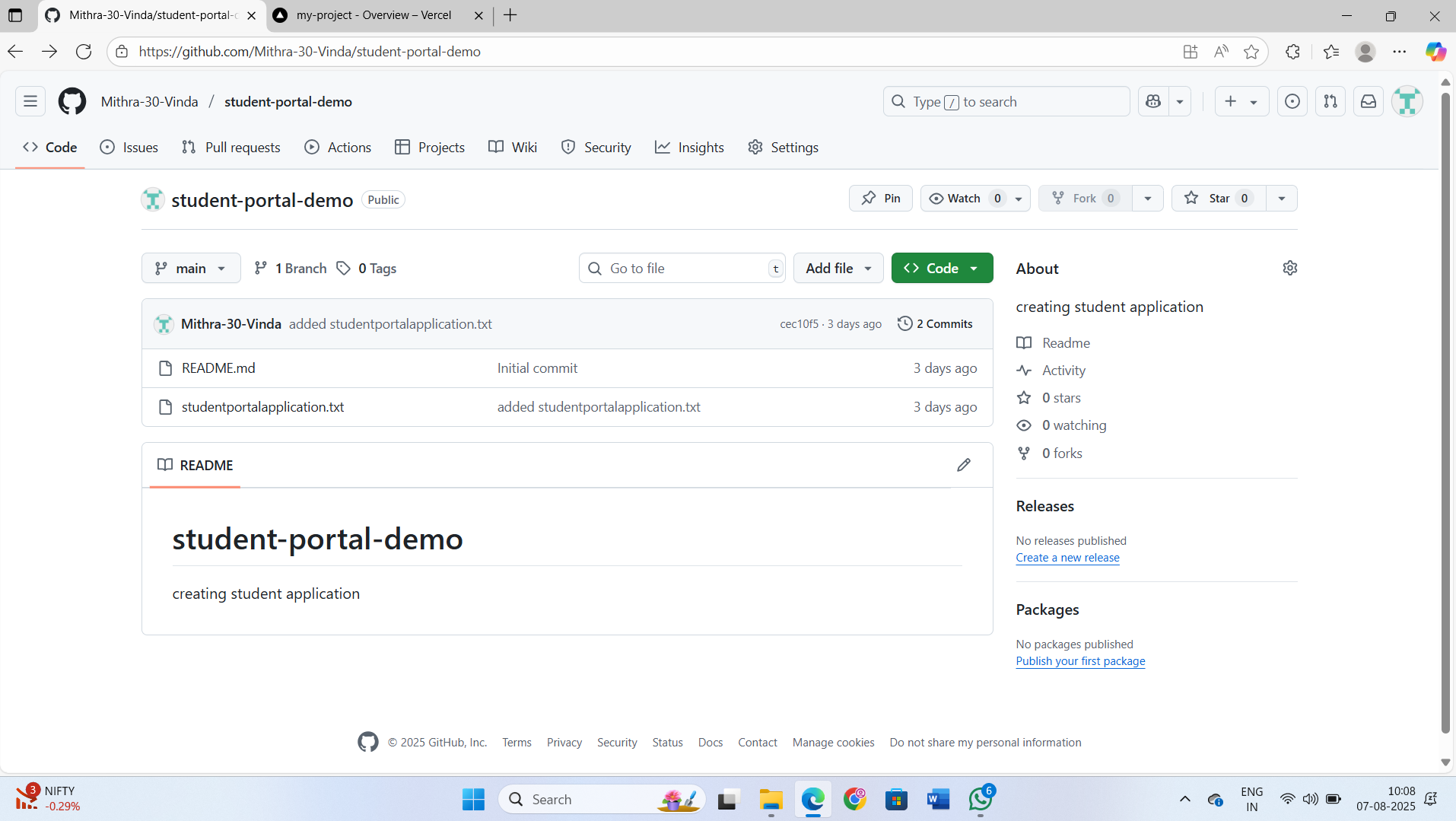


Repo link:

<https://github.com/Mithra-30-Vinda/My-first-repo/blob/main/about.txt>

**Experiment 3:**

Create a repo named student-portal-demo. Initialize it with a README file describing a basic idea for a student portal application. Share the link to your repository after completing the task.



Repo link:

<https://github.com/Mithra-30-Vinda/student-portal-demo>

**DevOps Visualizing:**

Bringing processes to life with diagrams.net

> DevOps involves multiple stages --like Plan, Develop, Build, Test, Release, Deploy, Operate, Monitor.

> Using diagrams.net helps students see the full flow clearly, instead of just reading theory.

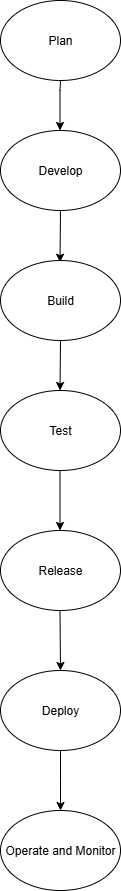
**Common Shapes and their Meanings:**

|  |  |  |
| --- | --- | --- |
| Shape | Name | Meaning/Use Case |
| Rectangle | Process/task | Represents a step in process |
| Diamond | Decision | Used for yes or no |
| Oval | Start/end | Represents start or end |
| Parallelogram | Input/Output | Denotes input or output operations |
| Document Shape | Document | Represents a file, report or documentation |
| Cylinder | Database/Storage | Used for database or storage |
| Folder Shape | Data set/Repository | Used to represent a collection of repositories |
| Cloud Shape | Cloud/Internet | Represents cloud services or external internet system |
| Arrows | Connector | Show the direction or flow between components |
| Cube/3D | Service/Micro Service | Represents a Standalone service |
| Gear shape | Operation or Automation tool | Indicates automated tools or processes |
| Person Shape | Actor/user | Represents a human |
| Circle | Connector node | For Off-page connections |
| Node/Sticky | Automation/Comments | Used to add explanations |

**Experiment 4:**

Design a DevOps Lifecycle Using diagrams.net Tool

<https://app.diagrams.net/> (also called Draw.io)



Repo link: <https://github.com/Mithra-30-Vinda/My-first-repo>

**Experiment 5:**

Design a DevOps lifecycle diagram for Student information system.

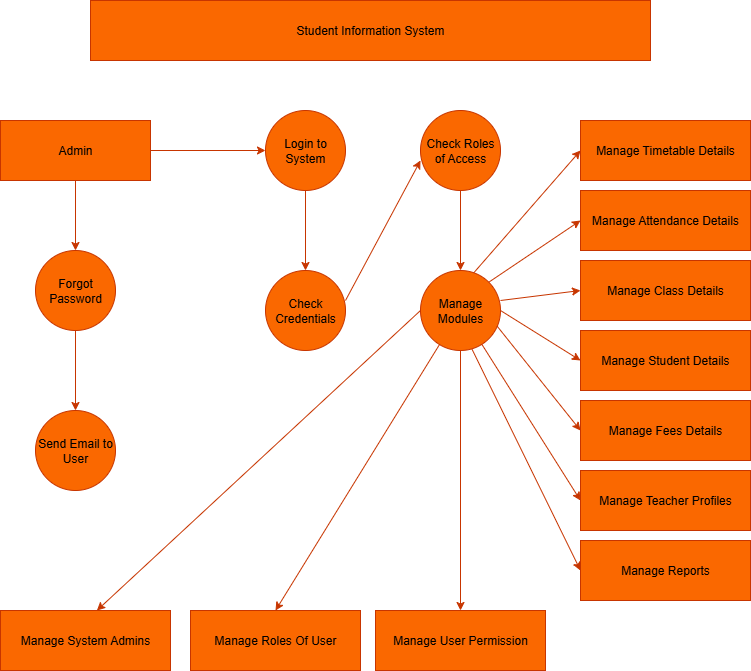


Diagram link: <https://github.com/Mithra-30-Vinda/My-first-repo>

**Experiment 6:**

Design a DevOps lifecycle diagram for Face recognition Attendance System.

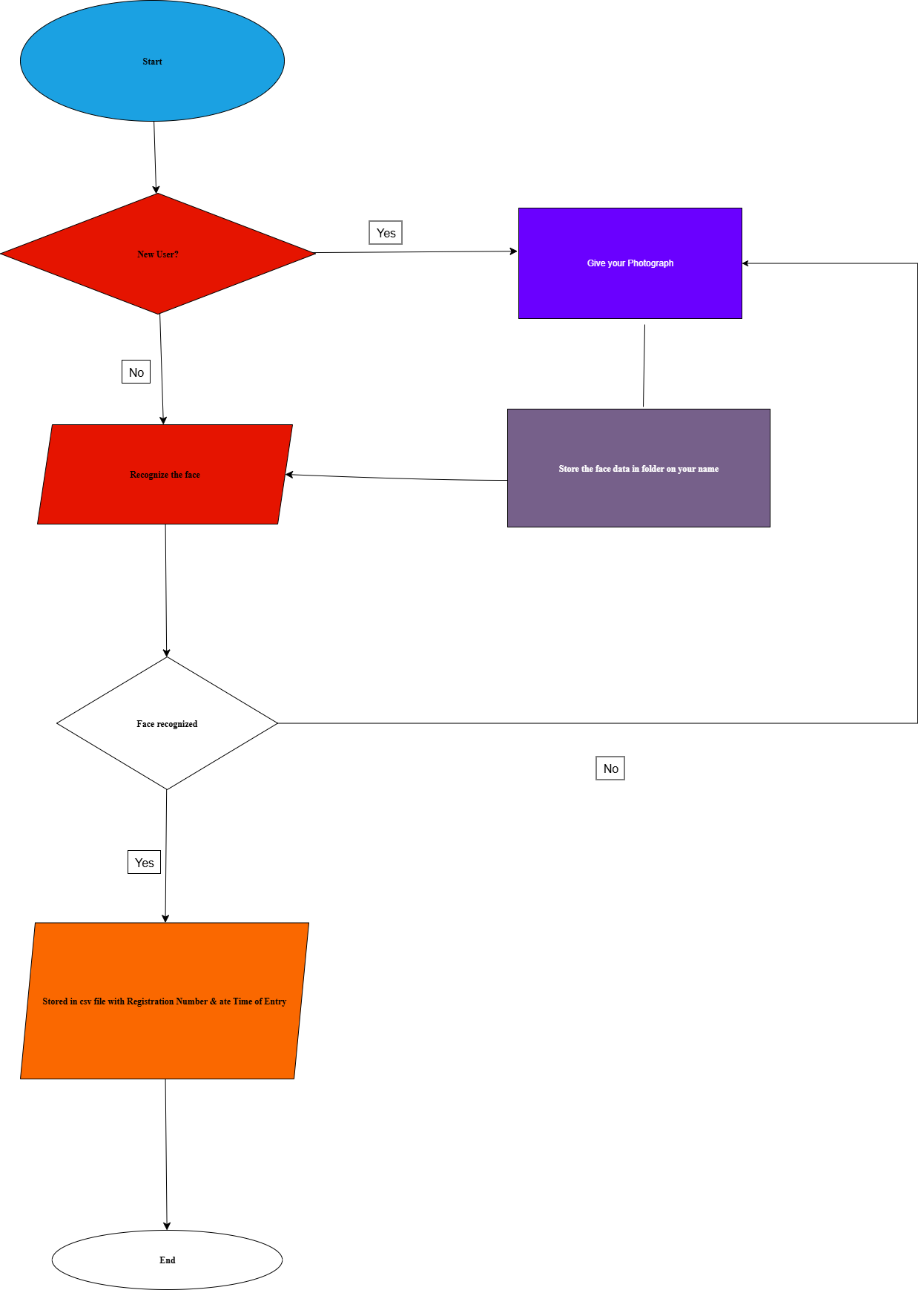


Diagram link

Food Delivery App like Zomato, Swiggy. : <https://github.com/Mithra-30-Vinda/My-first-repo>

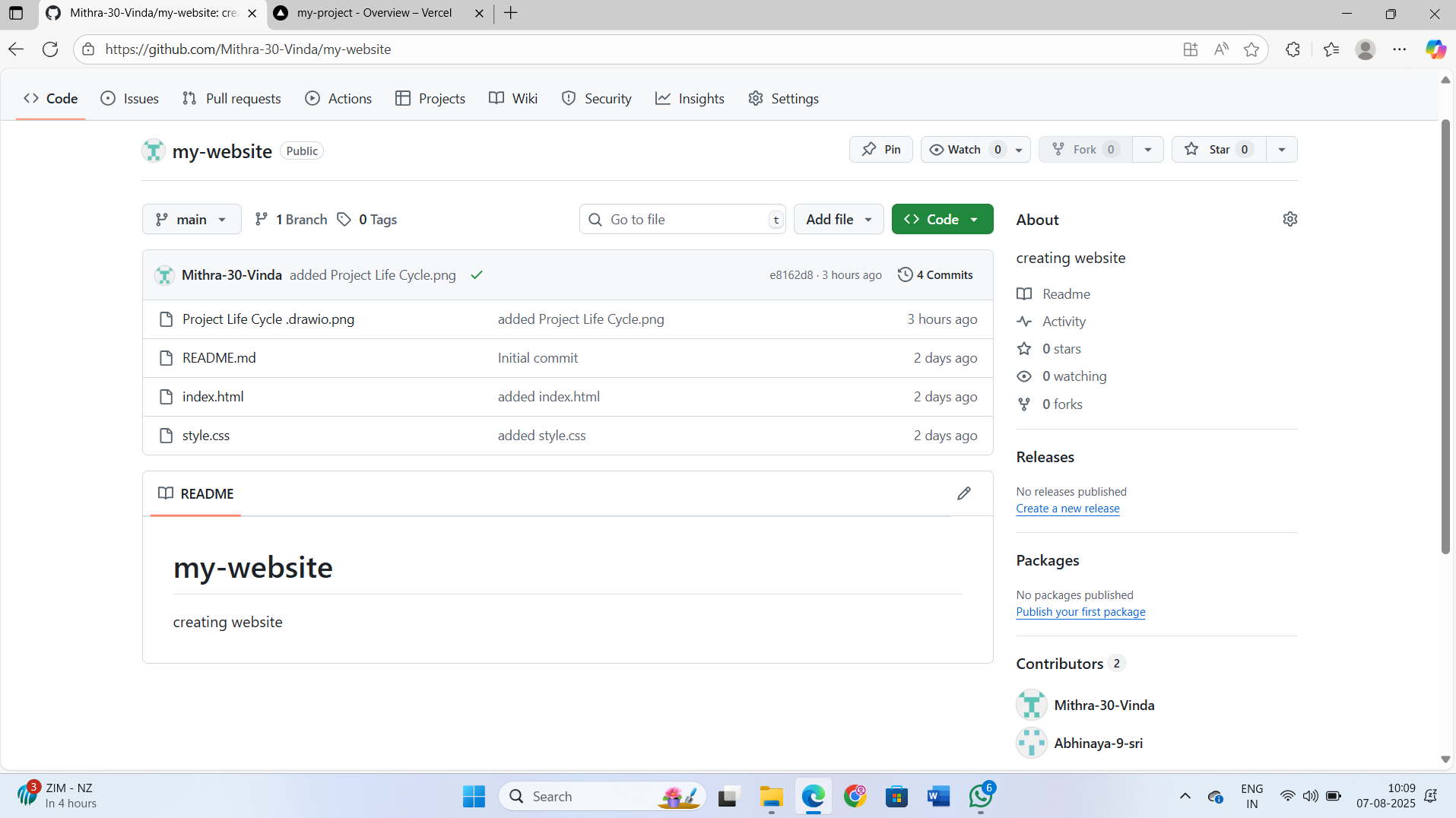
**Experiment 7:**



Diagram link : <https://github.com/Mithra-30-Vinda/My-first-repo>

**Experiment 8**:

Create your own web application.



Repo link:

<https://github.com/Mithra-30-Vinda/my-website>

Website link:

[my-website/index.html at main · Mithra-30-Vinda/my-website](https://github.com/Mithra-30-Vinda/my-website/blob/main/index.html)

**Experiment 9:**

DevOps Lifecycle for your web application.

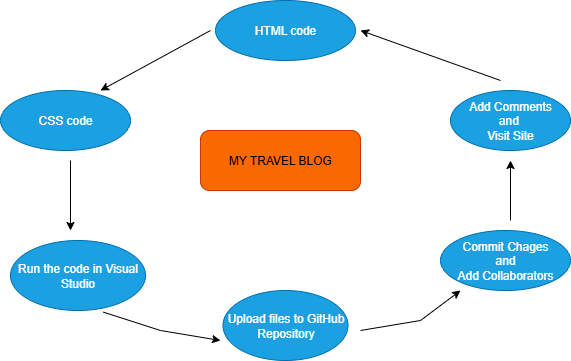


Diagram link: [my-website/Project Life Cycle .drawio.png at main · Mithra-30-Vinda/my-website](https://github.com/Mithra-30-Vinda/my-website/blob/main/Project%20Life%20Cycle%20.drawio.png)

**Docker:**

It is a tool that allows you to package and application with all dependencies into a container. So, it can run anywhere. For example, on your computer or server.

**Containers:**

Containers are light-weight, portable and self-contained s/w environments that include everything to run application, code, run time, system tools, libraries and settings. They ensure that application run recognition different environments whether on a developer's laptop or in production.

**Consistency:** Works on any machine.

**Speed:** Containers start quickly and use fewer system resources.

**Isolation:** Each container run separately avoiding conflicts.

**Scalability:** Easy to replicate and scale application.

**Virtual Machines vs Containers:**

|  |  |  |
| --- | --- | --- |
| Feature | Virtual Machines | Containers |
| Size | Heavy (GB's) | Light-weight (MB's) |
| Boot Time | Minutes | Seconds |
| Resource Usage | High (full OS per VM) | Low (shares OS kernel) |
| Isolation | Full OS Isolation | Process-level Isolation |
| Probability | Less Portable | Highly Portable |

**Why Docker in DevOps?**

Docker is the most popular containerization platform because:

>> It simplifies application packaging and distribution.

>> It integrates smoothly with CI/CD pipelines.

>> It improves developer productivity.

>> It enables easy testing and rollback of app versions.

**Experiment 10:**

Demonstrate how to upload a project to GitHub and deploy it using Vercel,

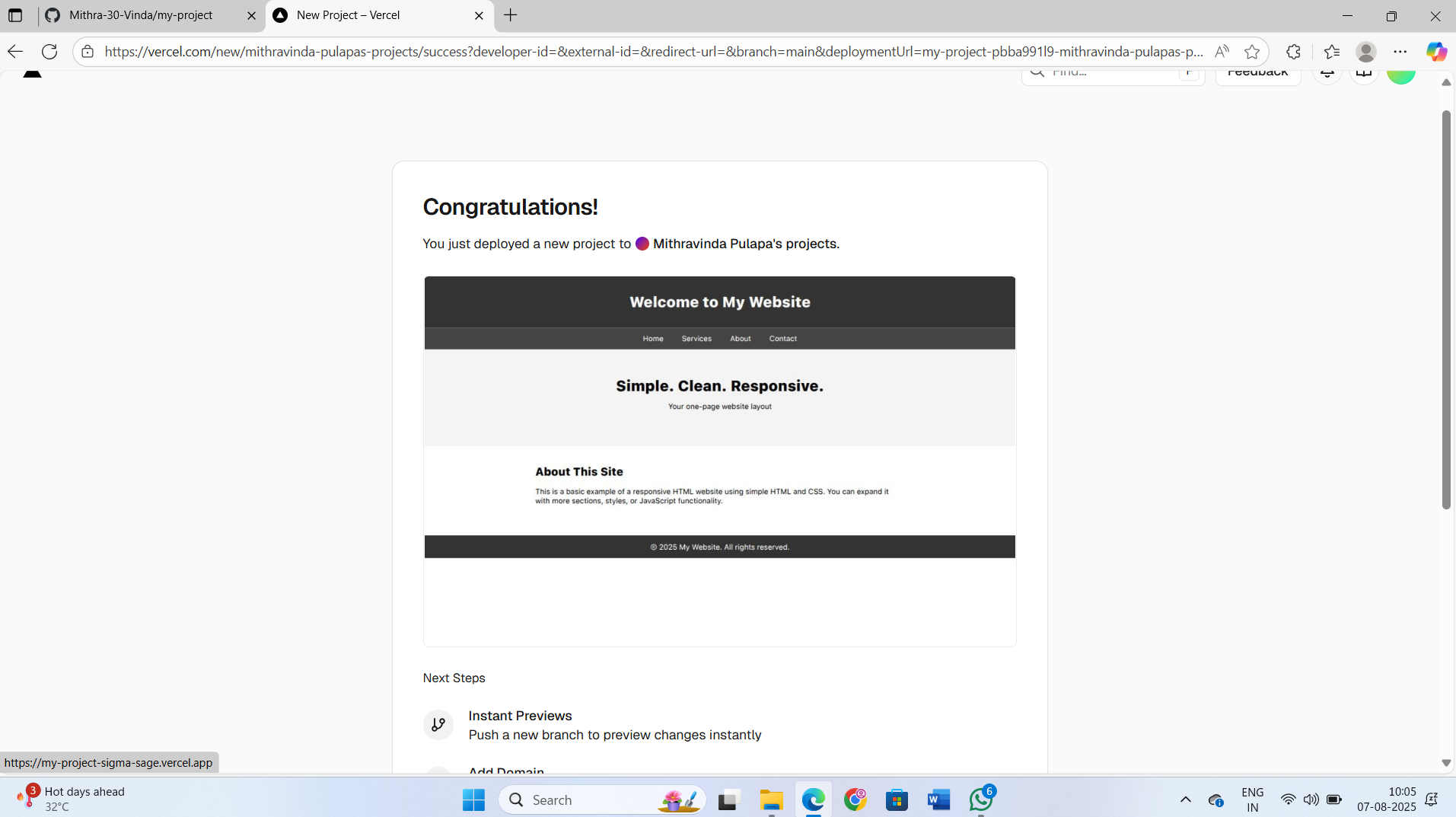
implementing a basic CI/CD workflow.

Step 1: Create New repository

Repo link: <https://github.com/Mithra-30-Vinda/my-project>

Step 2: Deploy on Vercel

<https://vercel.com/>



**Experiment 11:**

Design and develop a unique single-page web application based on your

own creative idea. Once completed, upload the project to GitHub and

deploy it using Vercel. The application must include at least one HTML file,

one CSS file, and basic JavaScript interactivity.

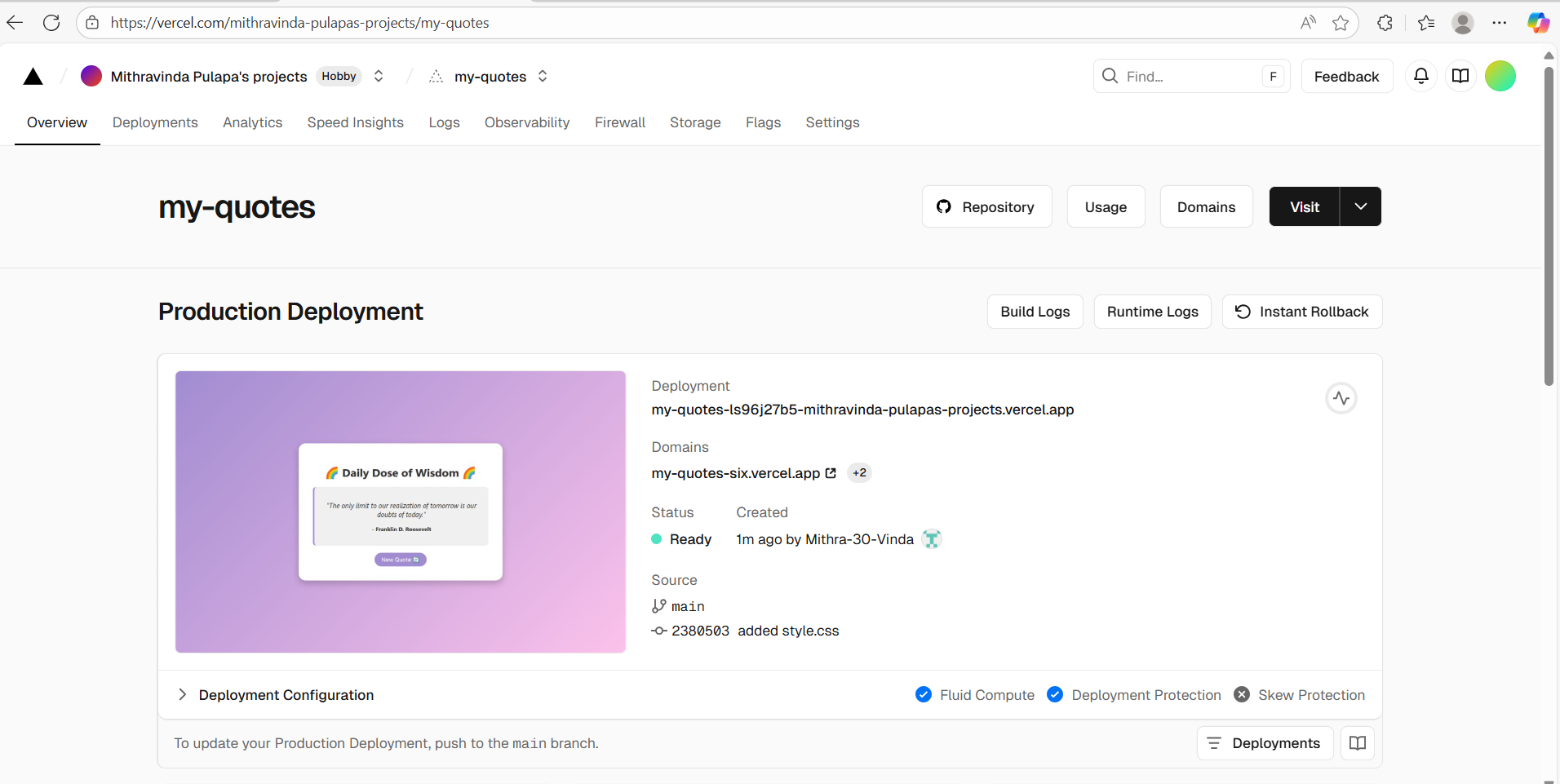
Submit the following:

Step 1: GitHub repository link

<https://github.com/Mithra-30-Vinda/my-quotes>

Step 2: Live deployed Vercel URL

Step 3: Screenshot of your Vercel dashboard showing successful deployment



Step 4: A short paragraph (3–5 lines) explaining the purpose of your web

Application

This code builds the structure of the admission form (fields like Name, Email, Phone, Course, DOB).

Includes a table where submitted data will be displayed.

Links external CSS and JavaScript files.

**Experiment 12:**

Create an HTML file that visually explains your previous experiment

DevOps lifecycle using diagrams, shapes, and labels. Ensure the file clearly

displays each stage of the DevOps lifecycle (Plan, Develop, Build, Test,

Release, Deploy, Operate, Monitor). Upload the HTML file to a GitHub

repository and deploy it using Vercel.

Submit the following:

1. GitHub repository link

2. Live Vercel deployment link

3. Screenshot of the live webpage showing the DevOps lifecycle diagram

4. Brief explanation (2–3 lines) of each stage within your HTML content

**Experiment 13:**

File Deletion from a GitHub Repository via Web Interface Steps:

1. Go to github.com and log in to your account.

2. Navigate to your repository where the file is uploaded.

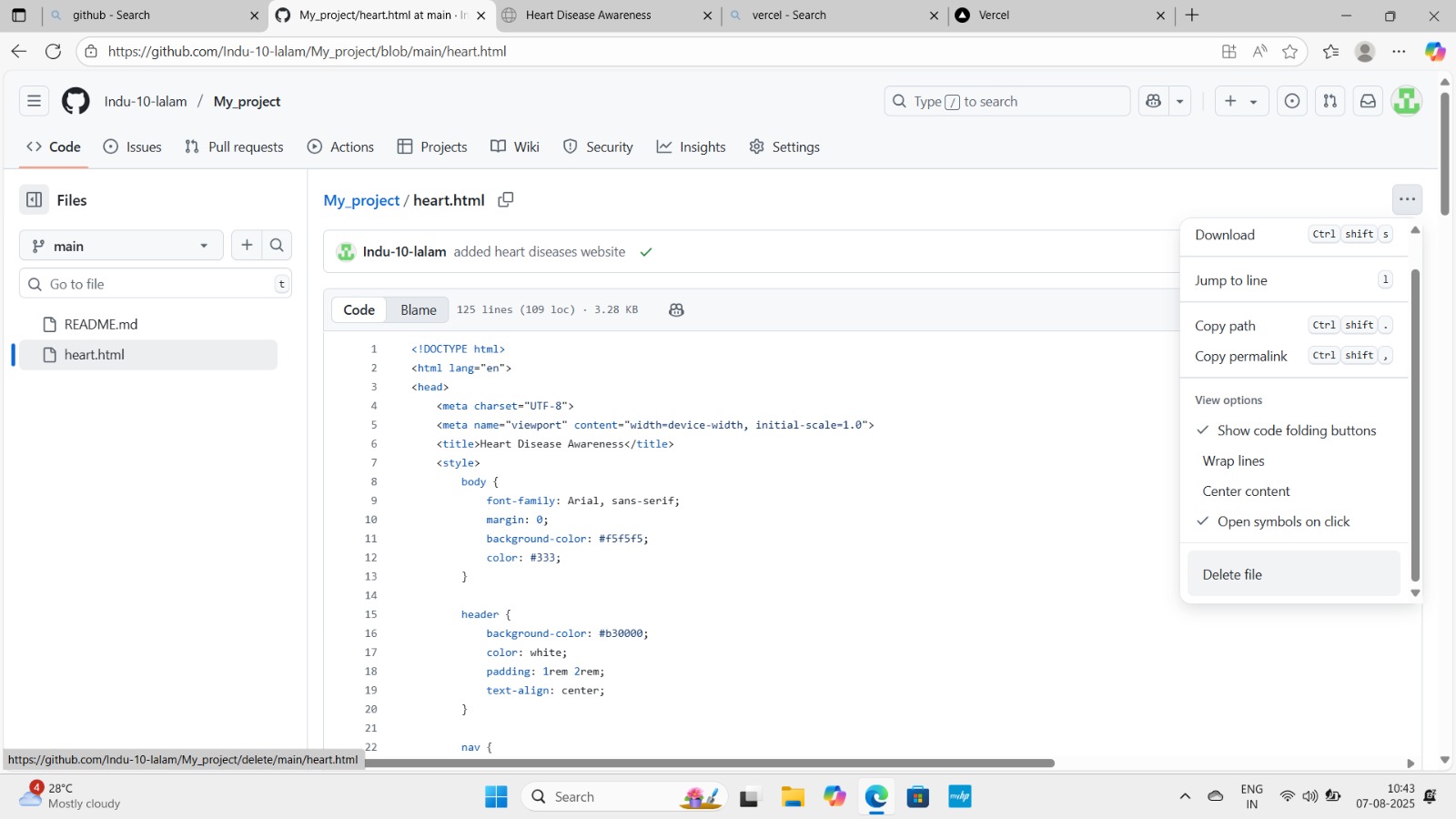
3. Locate the file you want to delete (e.g., index.html, file.txt, image.jpeg).

4. Click on the file name to open it.

5. On the top right of the file view, click the trash can icon labeled “Delete this file”.

6. On the next screen: GitHub will ask you to provide a commit message. (It can be something like "Deleted file.txt")

7. Scroll down and click “Commit changes” button.



**Experiment 14:**

Renaming a Repository on GitHub via Web Interface Steps:

1. Log in to your GitHub account at <https://github.com>.

2. Go to the repository you want to rename.

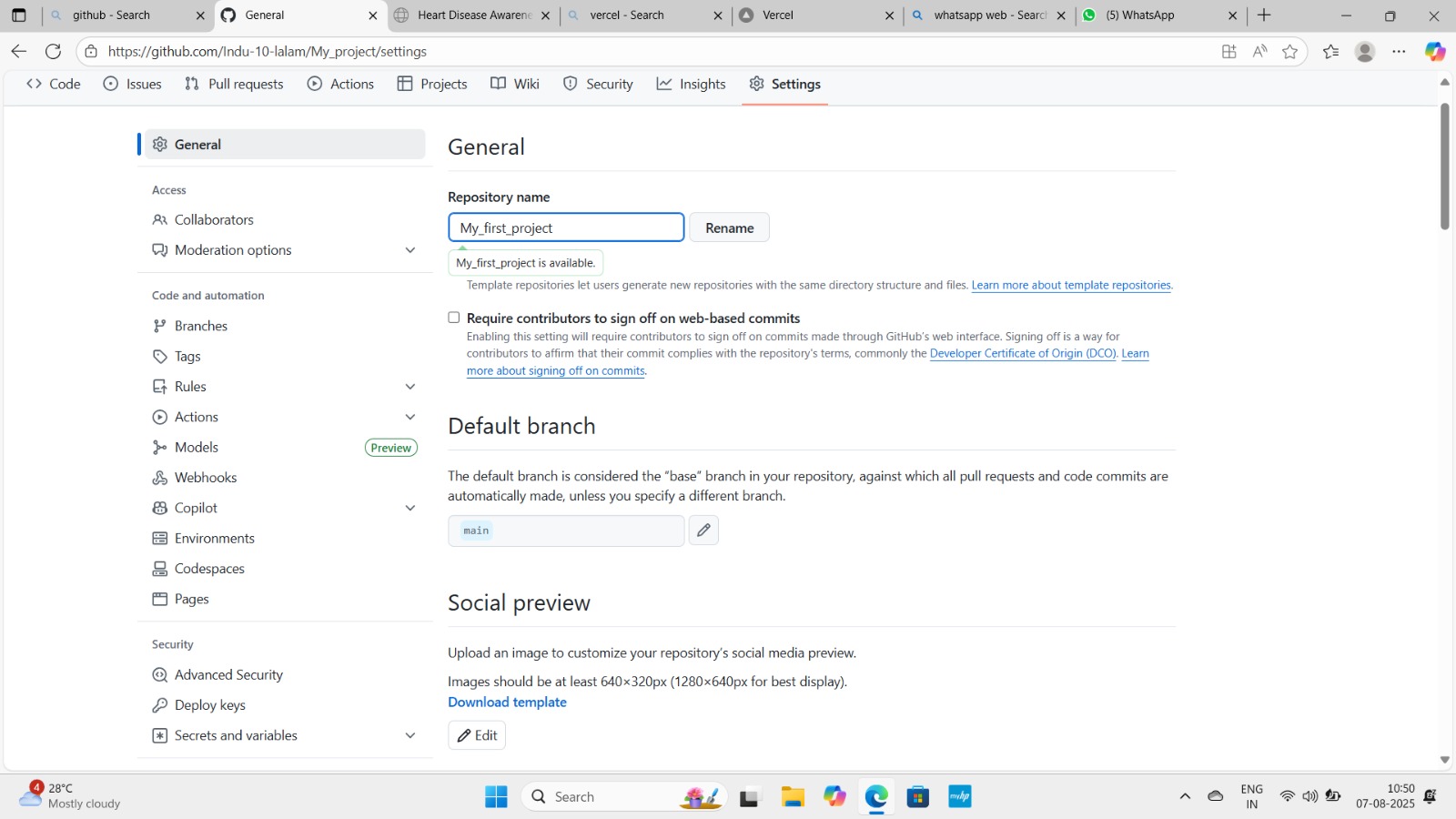
3. On the top menu bar, click on the “Settings” tab of the repository.

4. Scroll down to the “Repository name” section under the General settings: You’ll see the current name and an editable text box.

5. Click inside the box, type the new repository name.

6. After entering the new name, click the “Rename” button (usually blue colored).

7. GitHub will update the repository name immediately**.**



**Experiment 15 :**

Deletion of an Existing Repository on GitHub via Web Interface Steps:

1. Log in to your GitHub account at <https://github.com>.

2. Go to the repository you want to delete.

3. On the top menu bar, click on the “Settings” tab of the repository.

4. Scroll all the way down to the "Danger Zone" section.

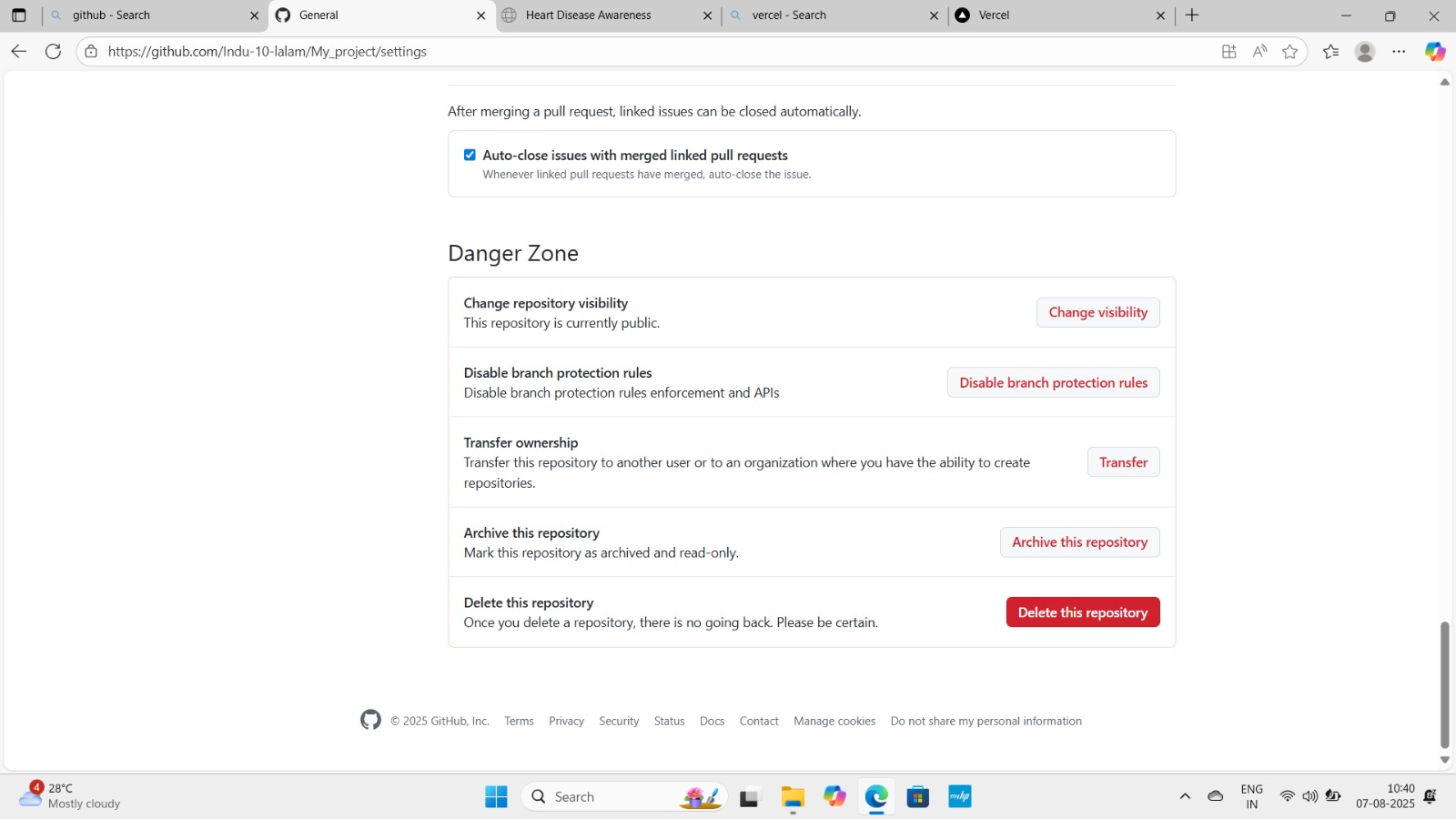
5. Look for the option “Delete this repository”.

6. Click the “Delete this repository” button (it will be red).

7. A confirmation popup will appear. GitHub will ask you to: Type the full repository name (e.g., yourusername/repository-name) to confirm. This is to prevent accidental deletion.

8. After typing the repository name, click “I understand the consequences, delete this repository”.

9. GitHub will permanently delete the repository and redirect you to your profile.



**Note :** Deleting a repository is **permanent.** Once deleted, all issues, pull requests, code, and wiki data will be lost unless backed up.