**DEVOPS IA2: IMPLEMENTATION OF CI/CD PIPELINE**

**Batch: LY-IT-B2**  **Tool Used: Buddy**

**Team Members:**

Dhruv Doshi – 1814002

Bhavik Bhatt – 1814007

Piyush Chavda – 1814010

Labdhi Jain – 1814015

Muskaan Nandu – 1814020

Kunj Gala – 1814020

1. **INTRODUCTION**

Buddy is a web-based and self-hosted continuous integration and delivery software for Git developers that can be used to build, test and deploy web sites and applications with code from GitHub, Bitbucket and GitLab.Configuration is performed by arranging predefined actions into sequences called pipelines. Pipelines can be triggered automatically on push to branch, manually, or recurrently. Actions include Docker-based builds, deployment to FTP/SFTP and IaaS services, delivery to version control, SSH scripts, website monitoring and conditional notifications. Contrary to other CI tools like Jenkins or Travis CI, Buddy does not use YAML files to describe the process, although the company stated support for .yml files is currently in works.

For the IA, we have developed a PHP website that is tested using PHP command line executions.

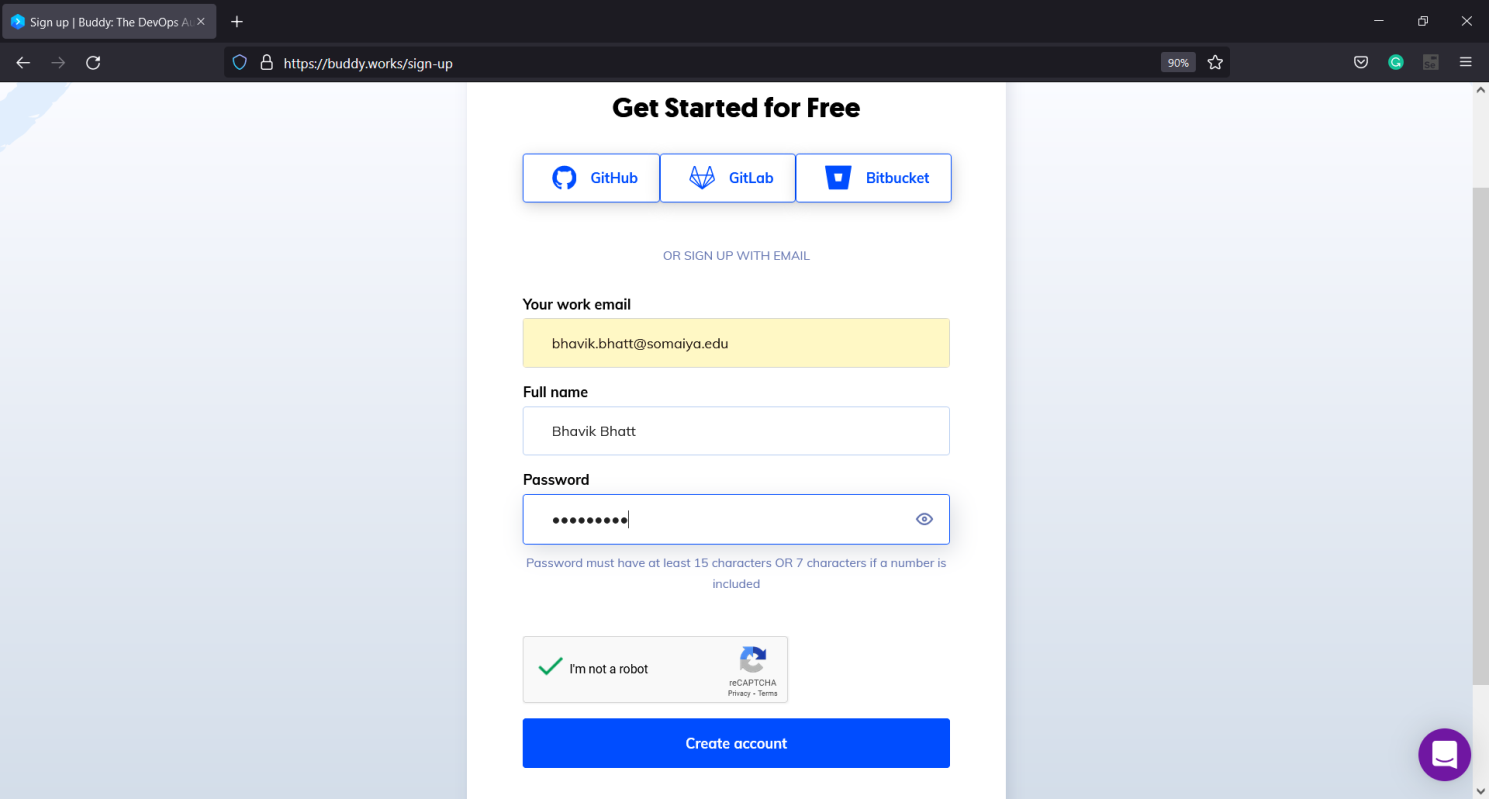
Compilation: Visual Studio Code

Version Control: Git and GitHub

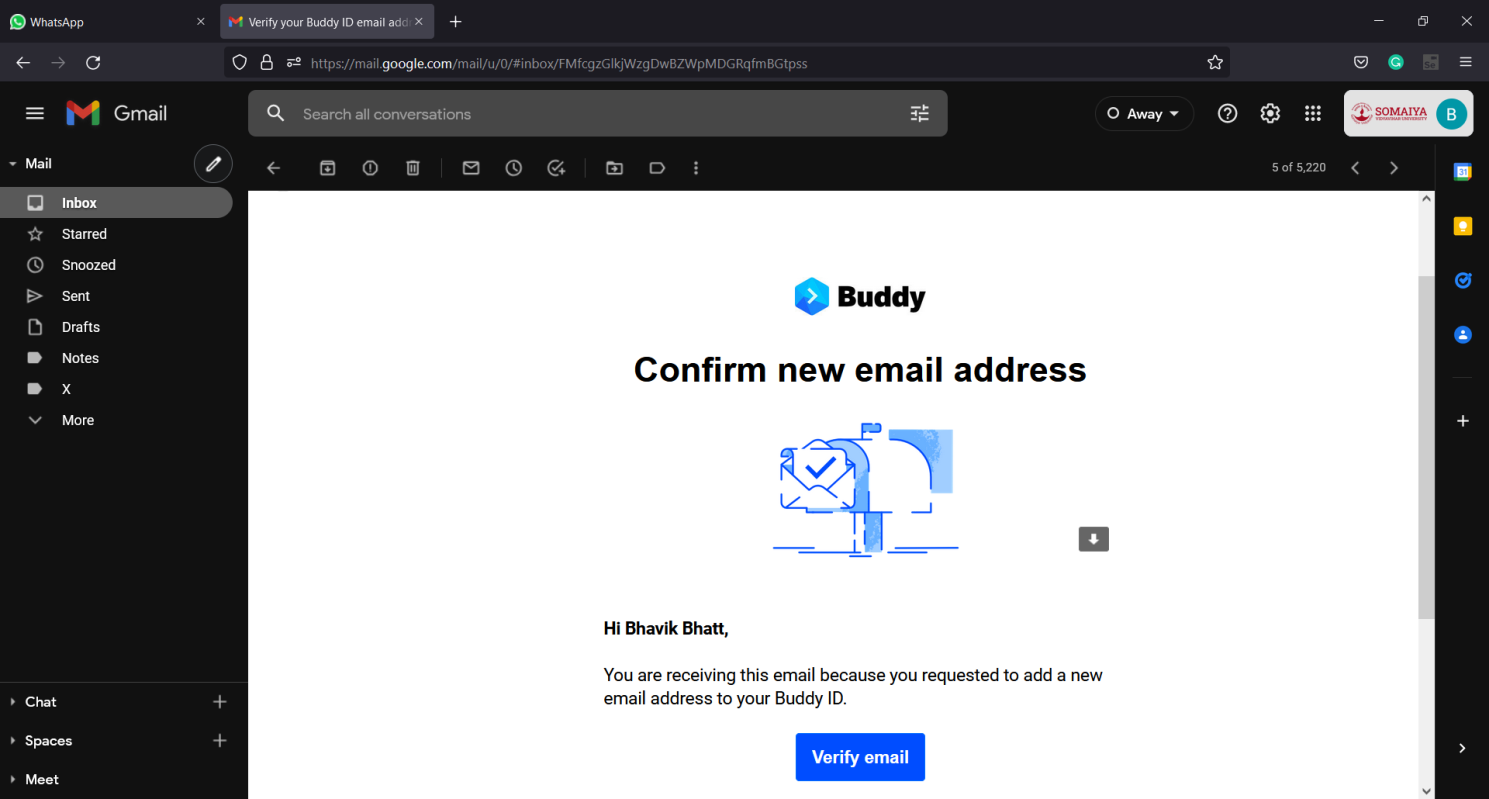
Deployment: GitHub Pages

1. **GETTING STARTED WITH BUDDY**

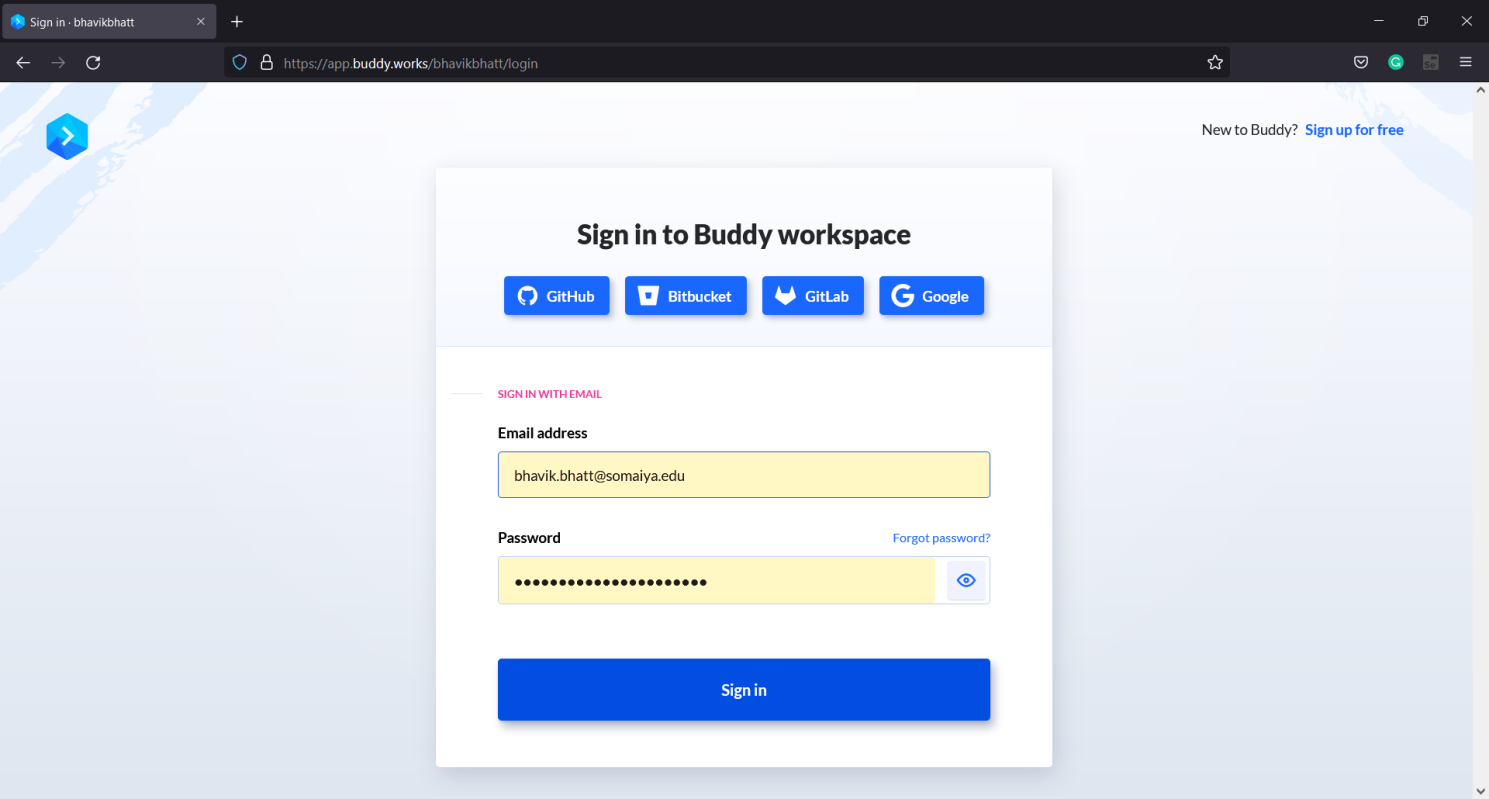
* Browsing to official Buddy website and signing up for free using email account: <https://buddy.works/>

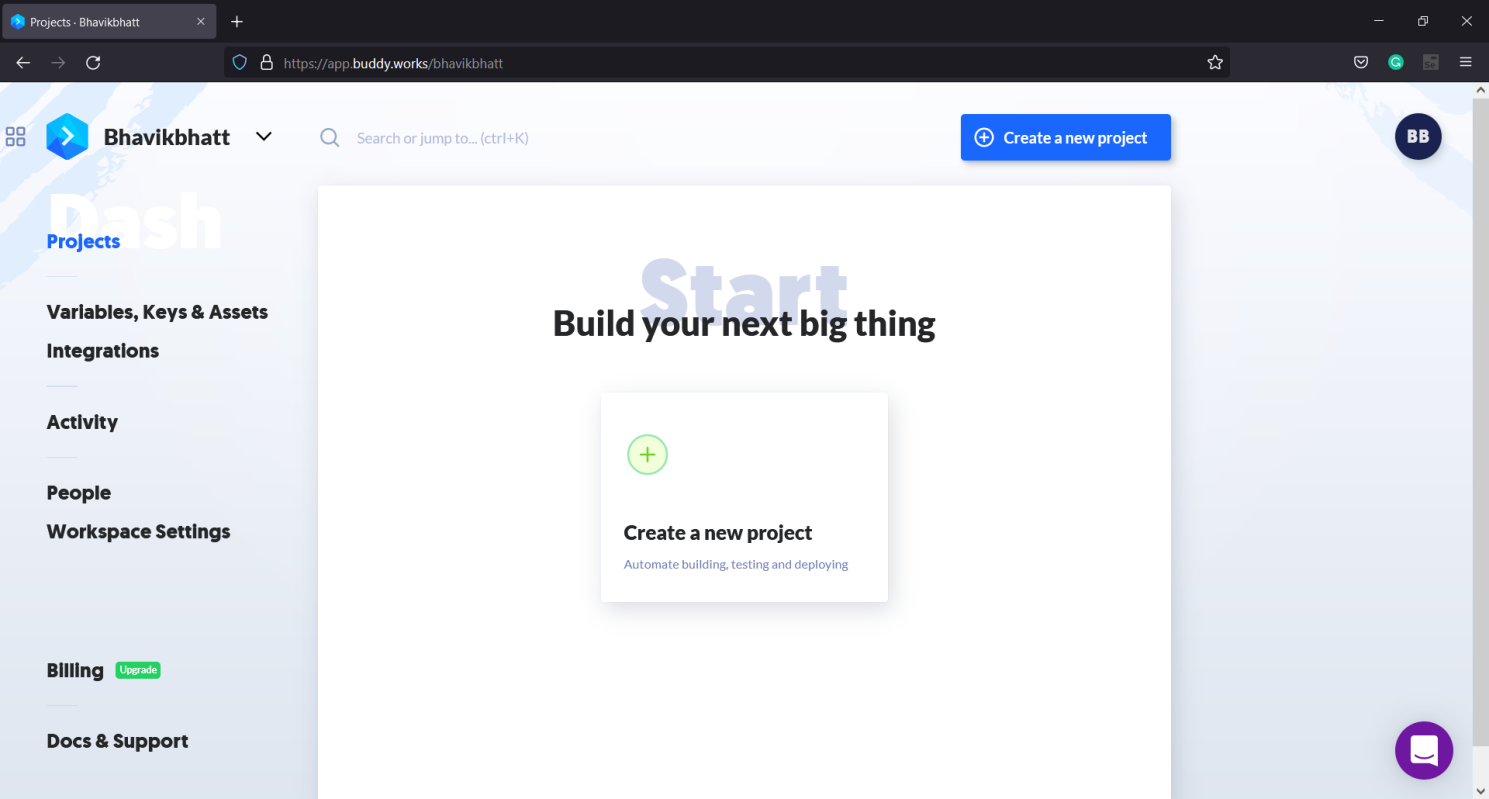


* Verifying the Email



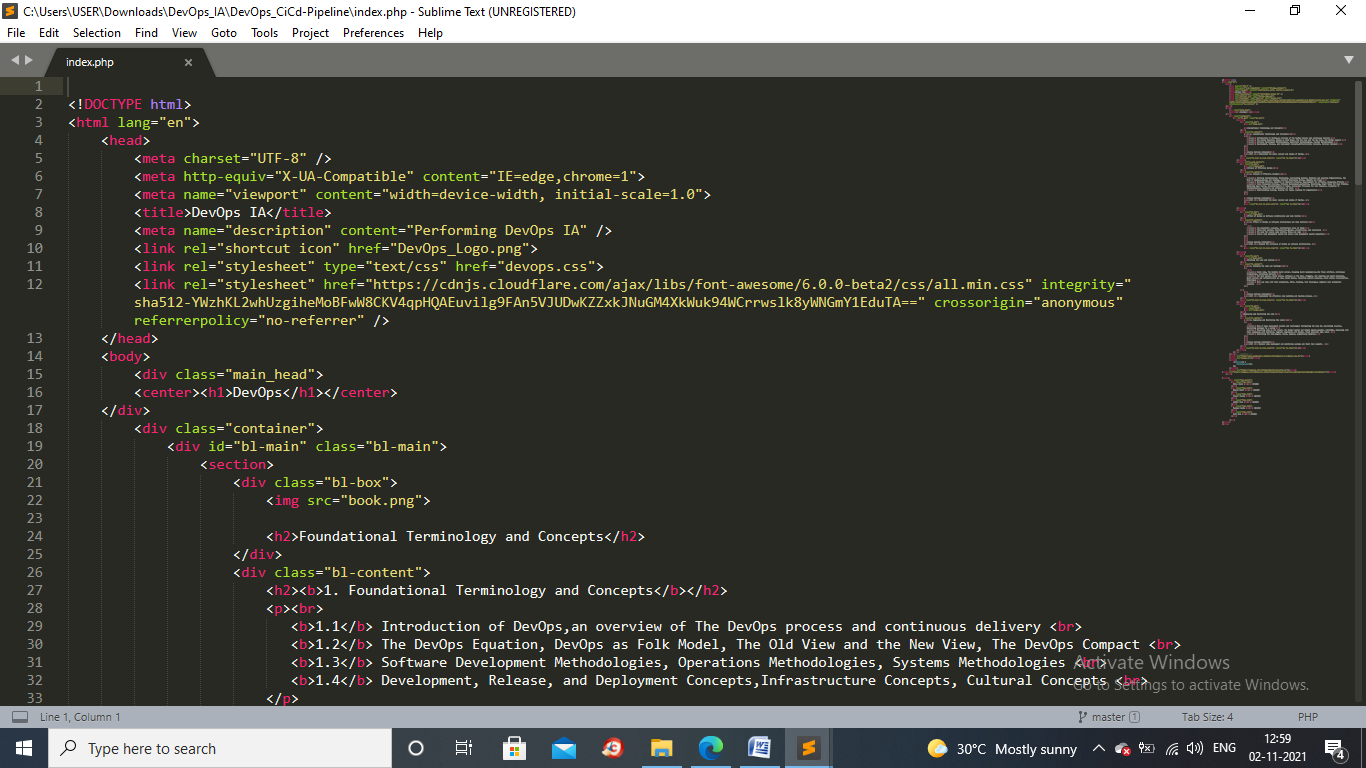
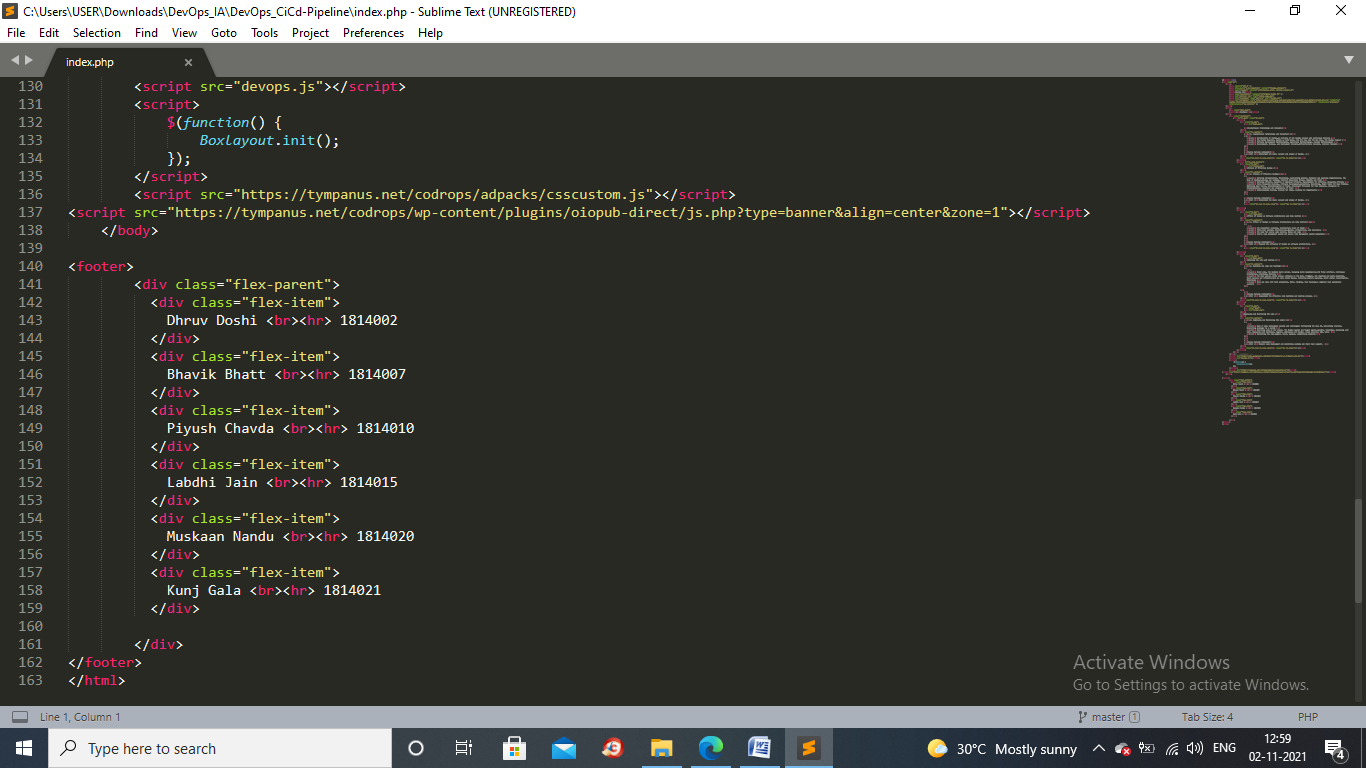
* Logging into the account once verified



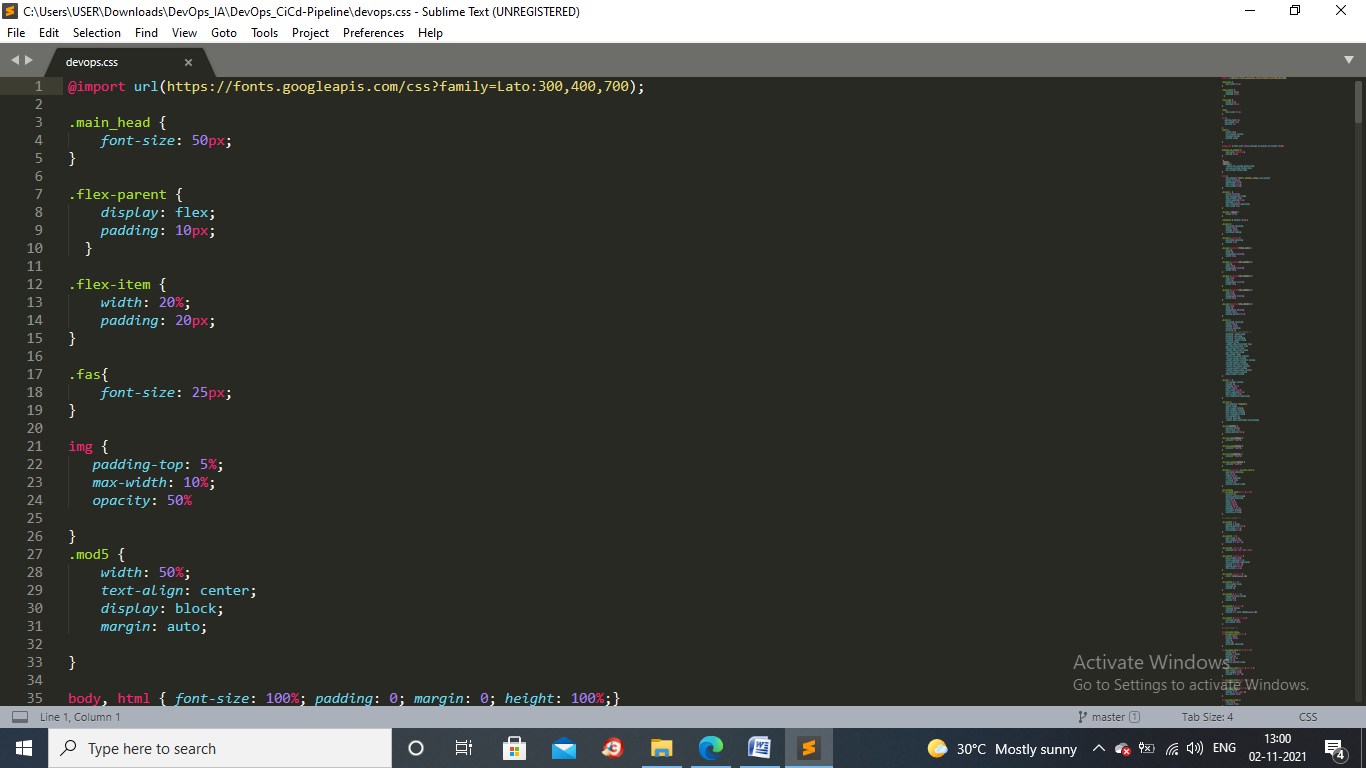
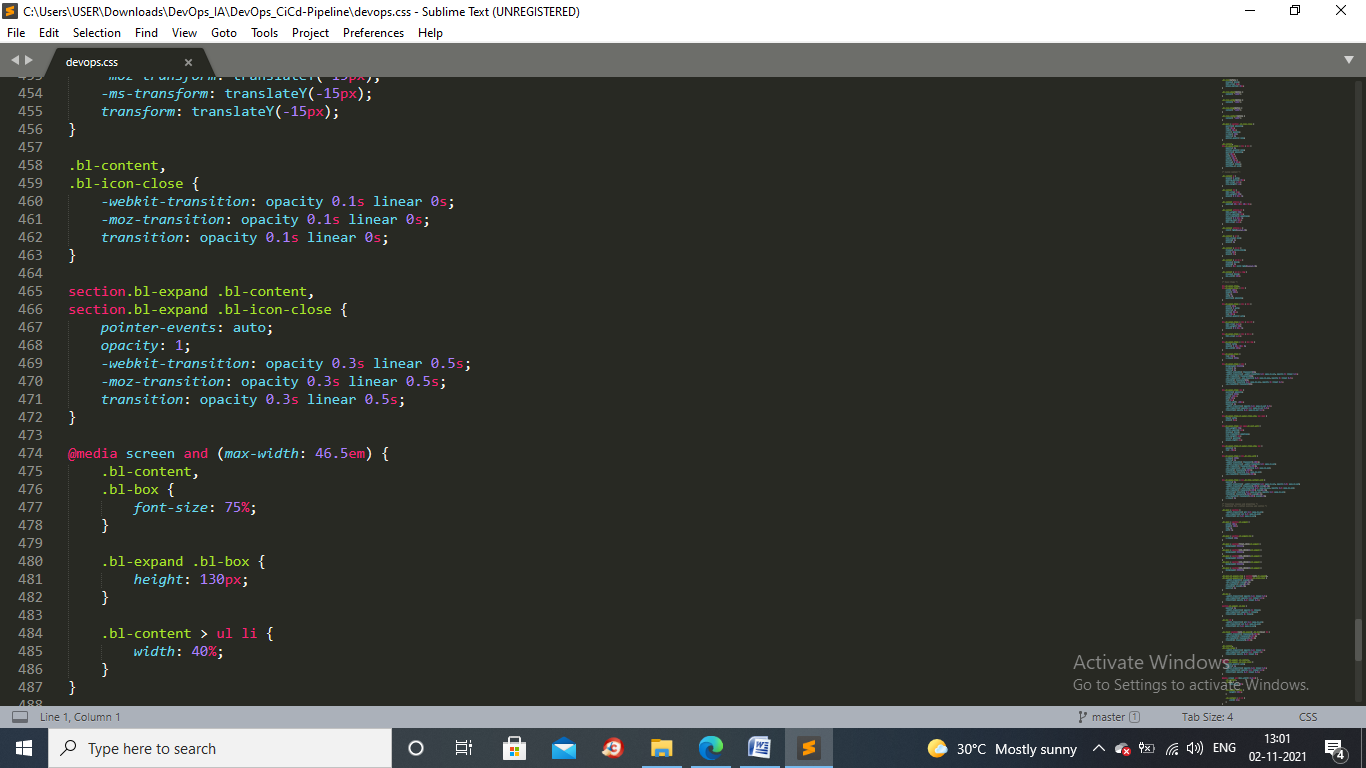


1. **CODE IMPLEMENTATION FOR WEBSITE**

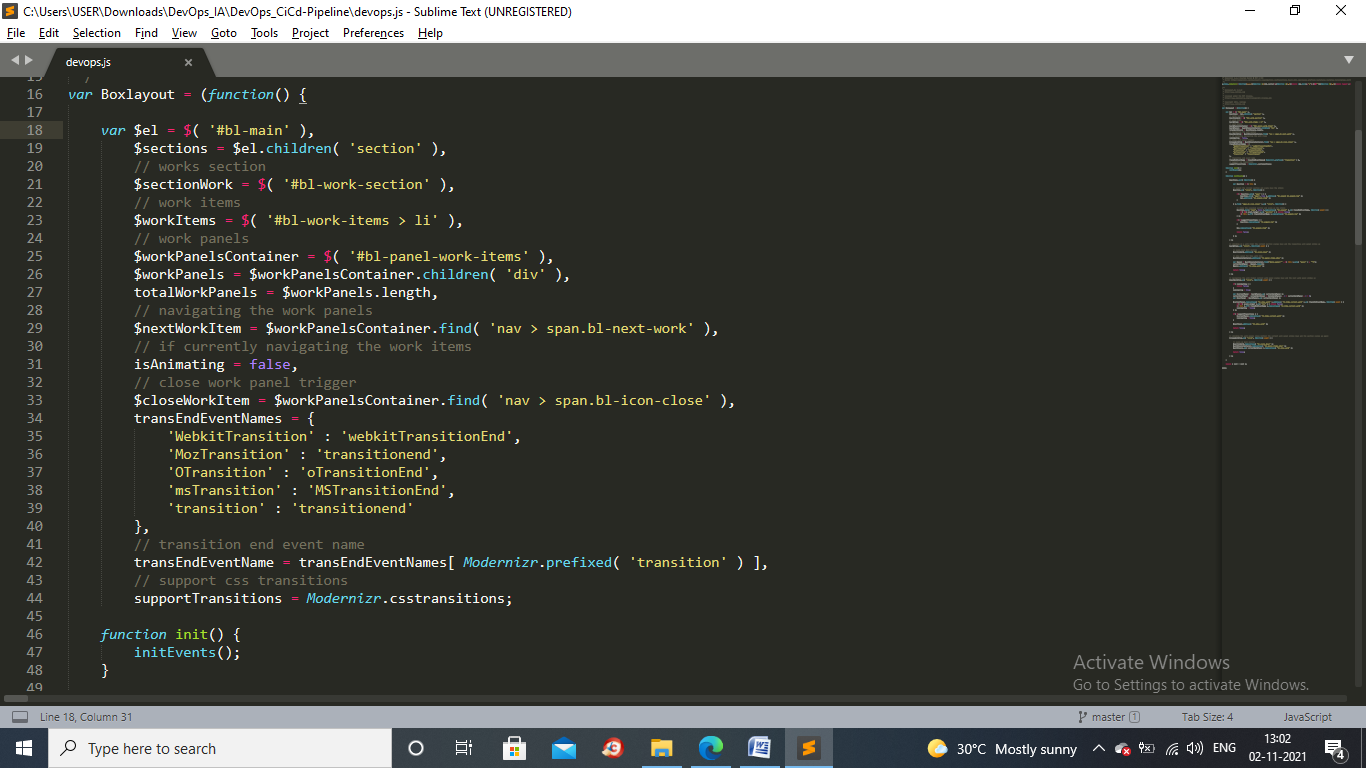
**Index.html File:**

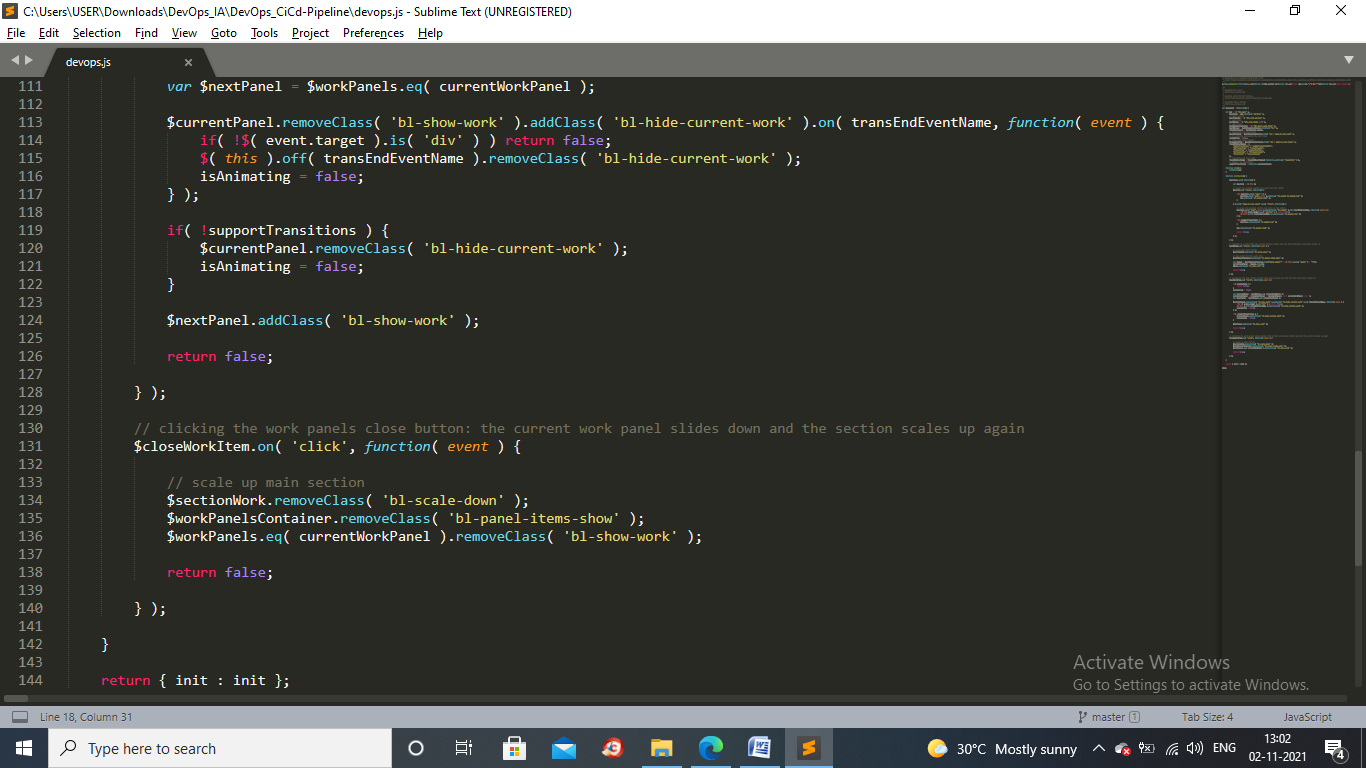
 

**Devops.css File:**

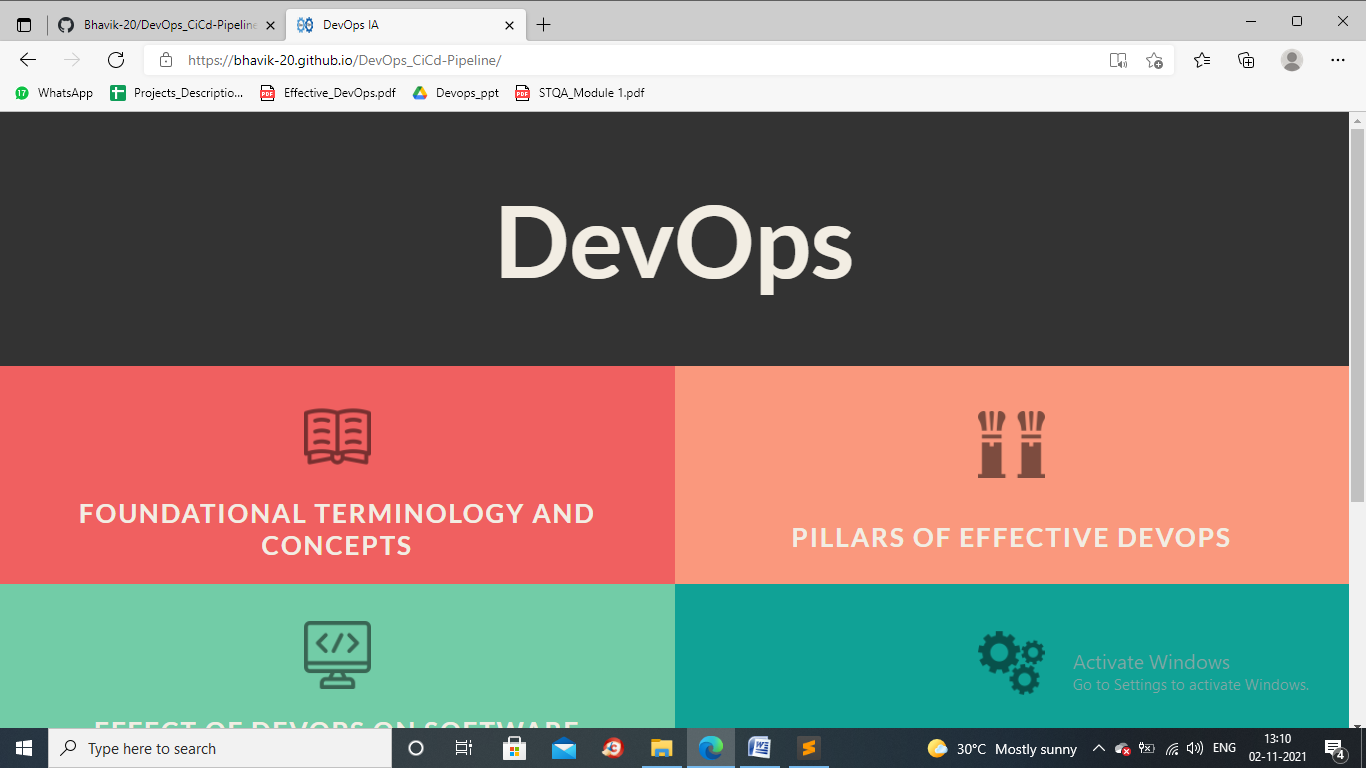
 

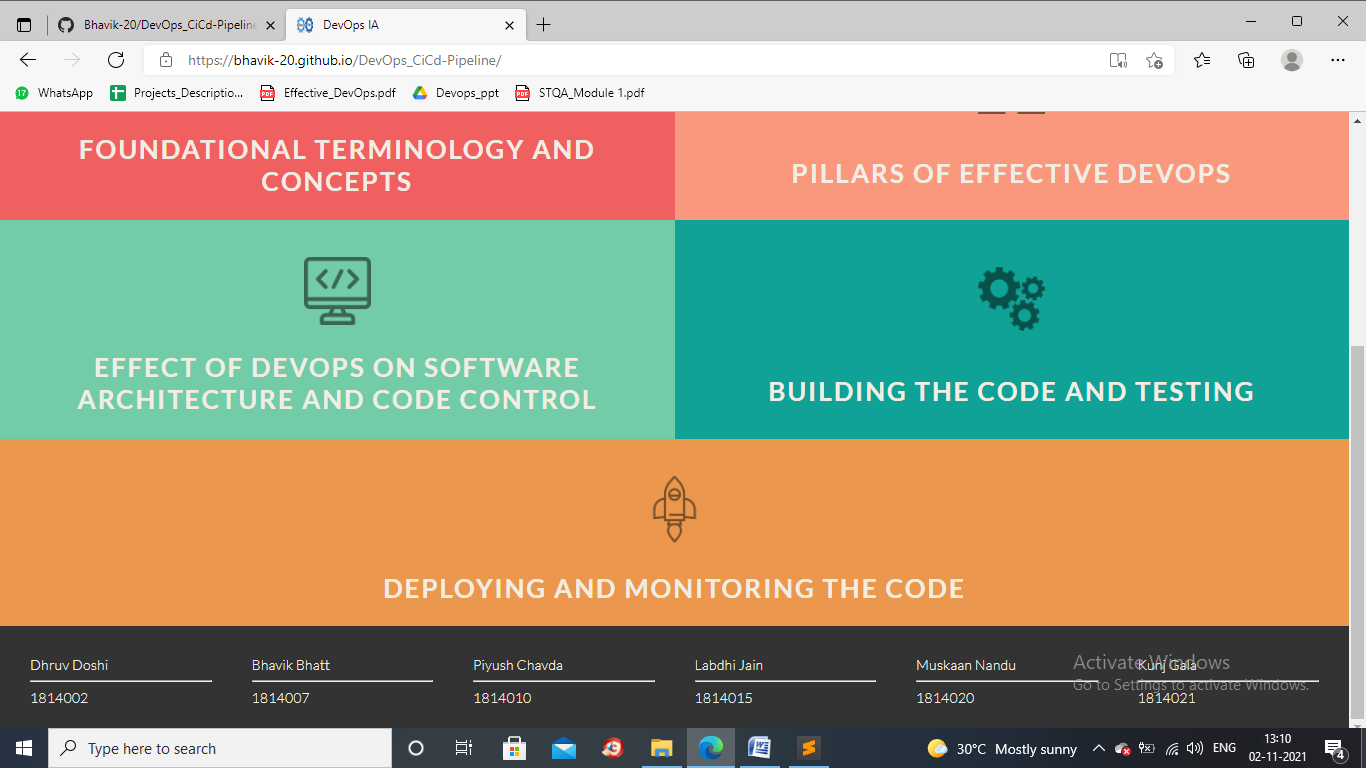
**Devops.js File:**

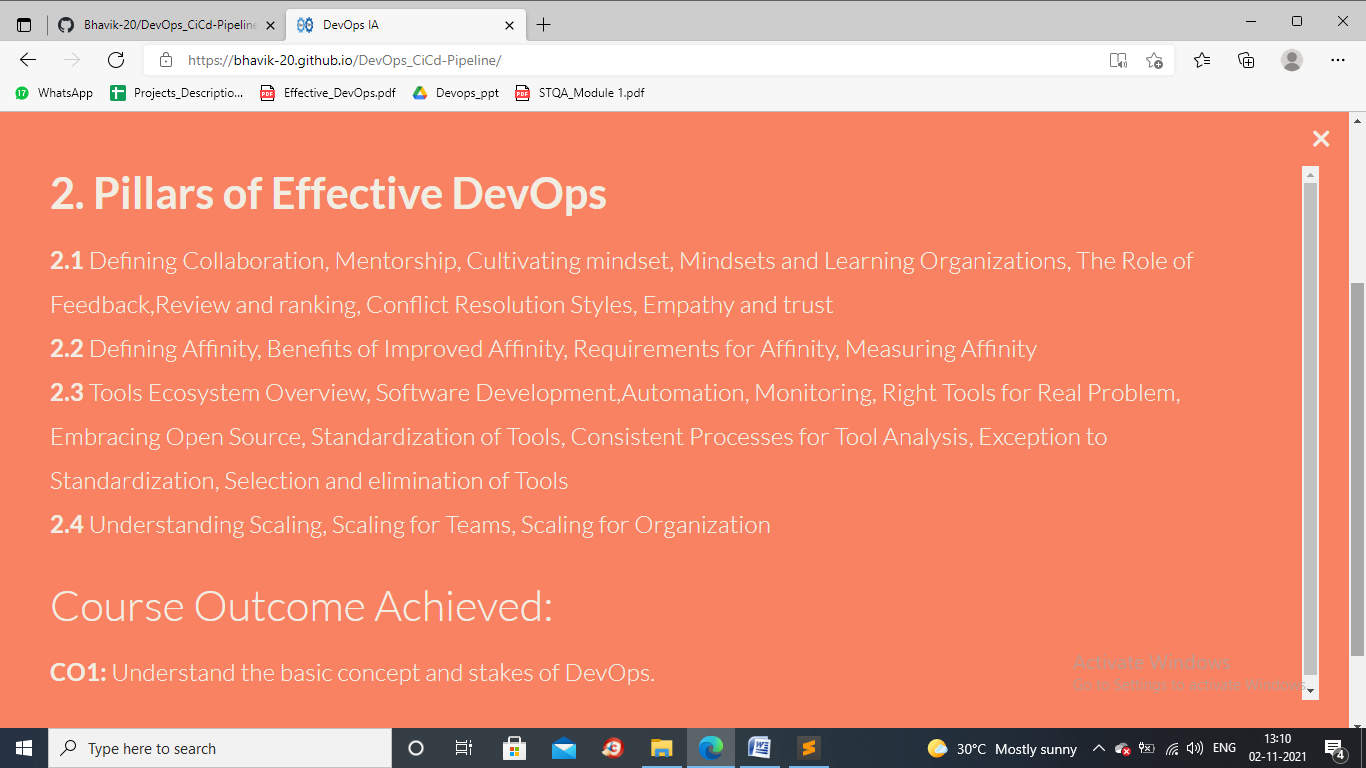




**Developed Website:**

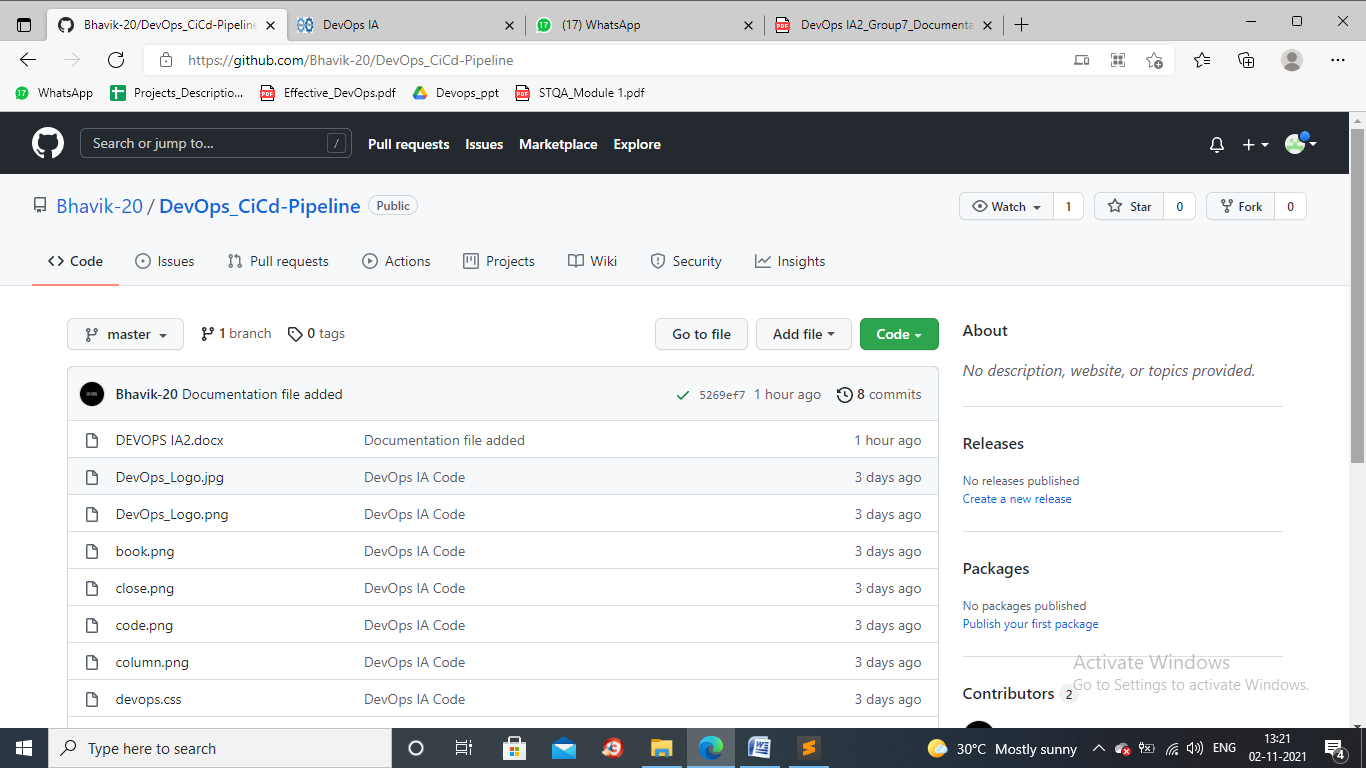




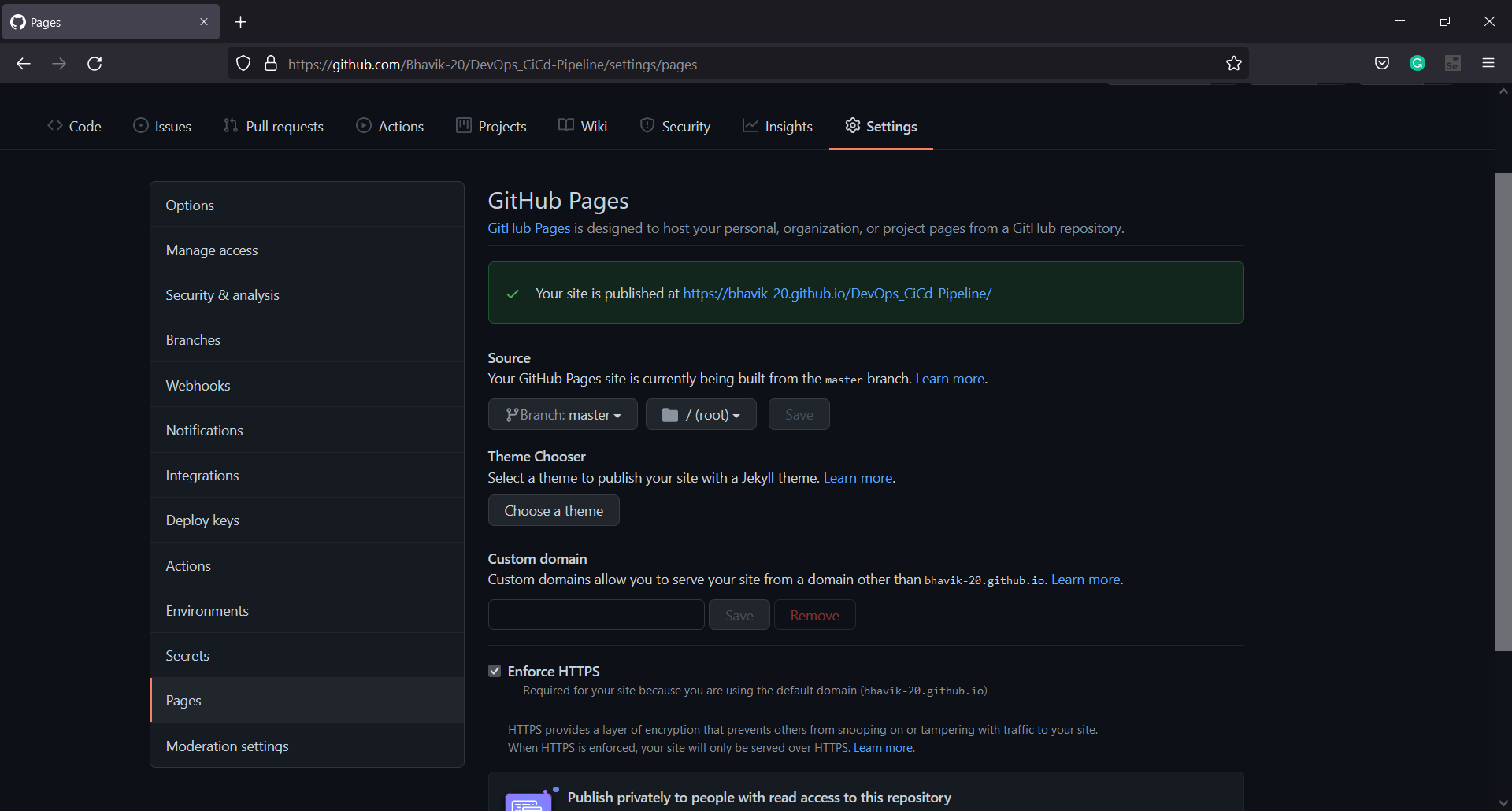


**Created Repository named DevOps\_CiCd-Pipeline and pushed code to it :**

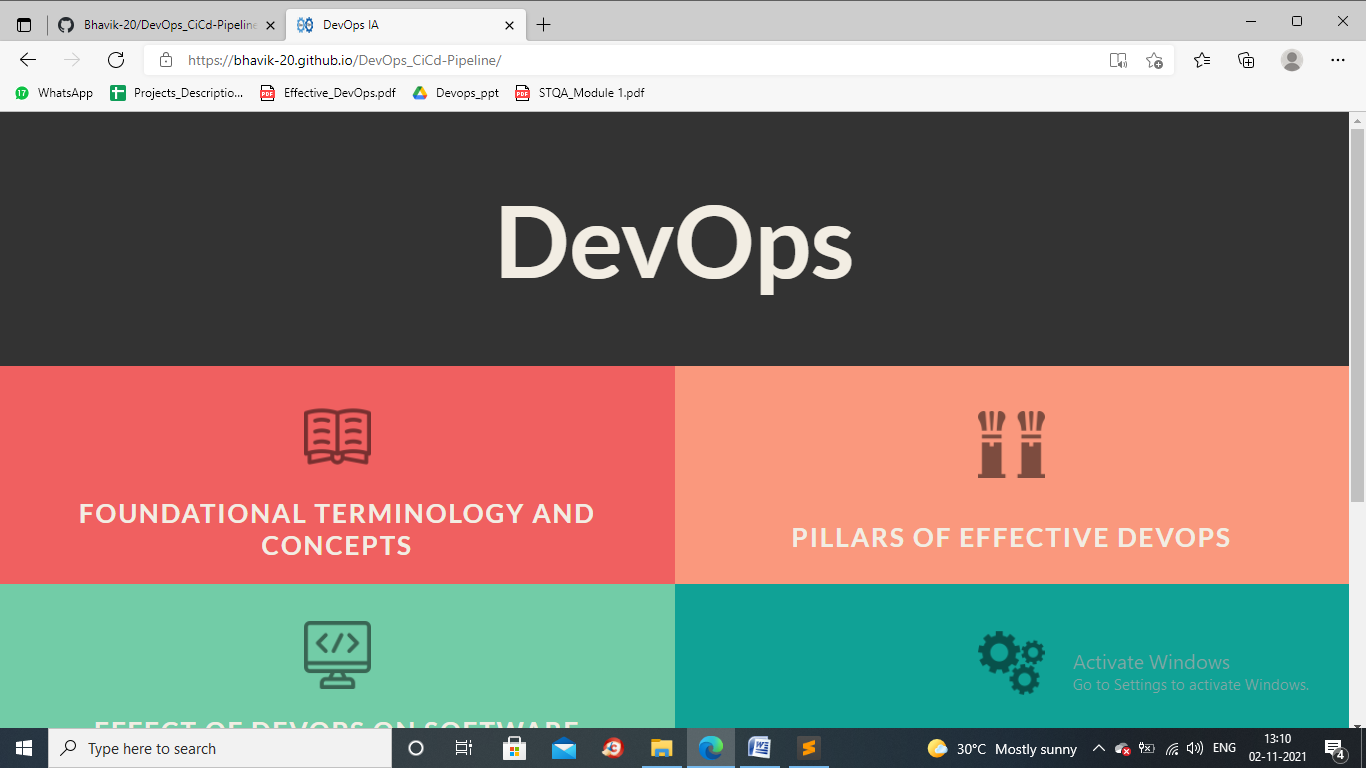
**Github repository :**



**Deployed the website on github pages:**

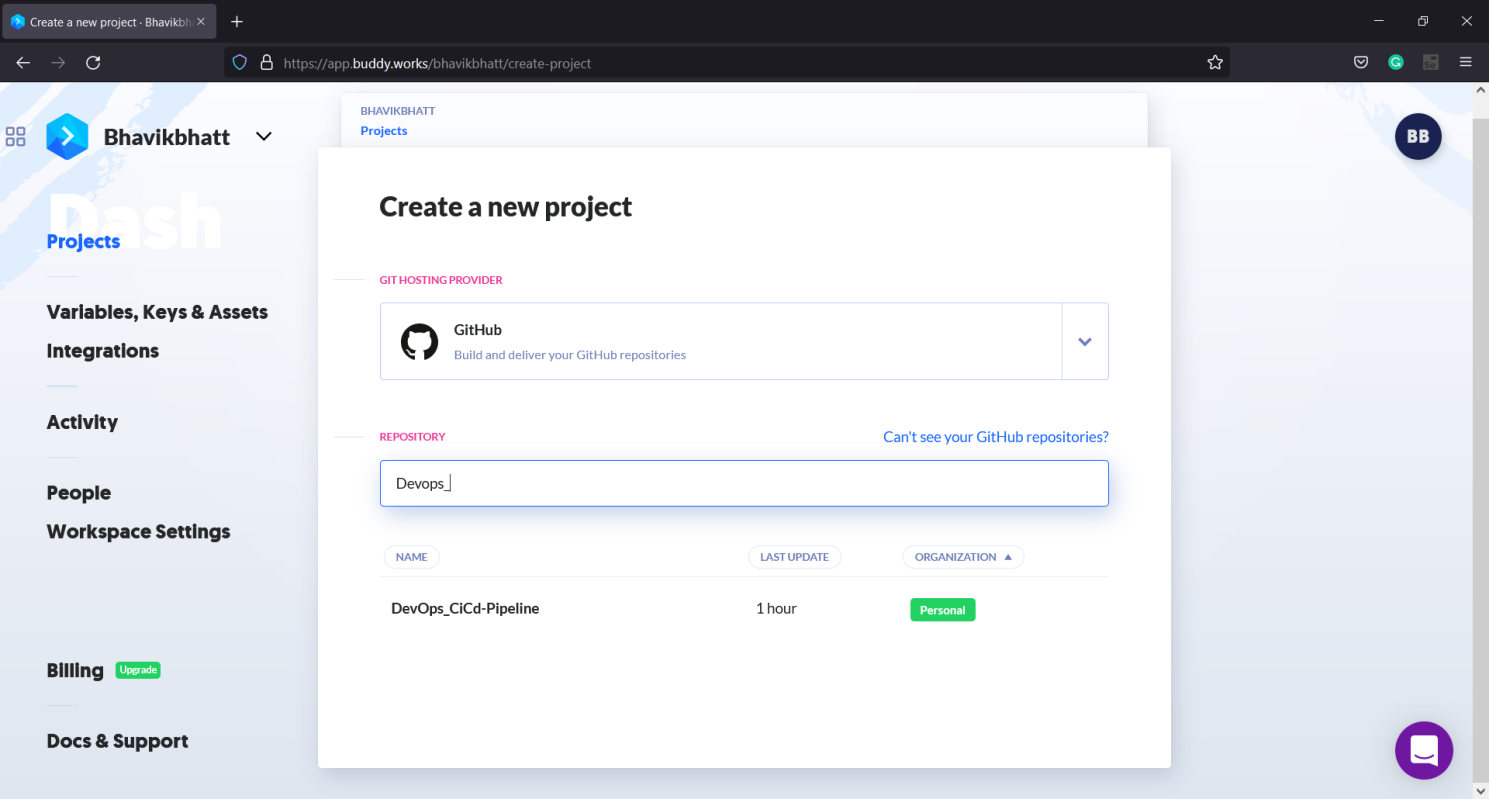


**Deployed website:**

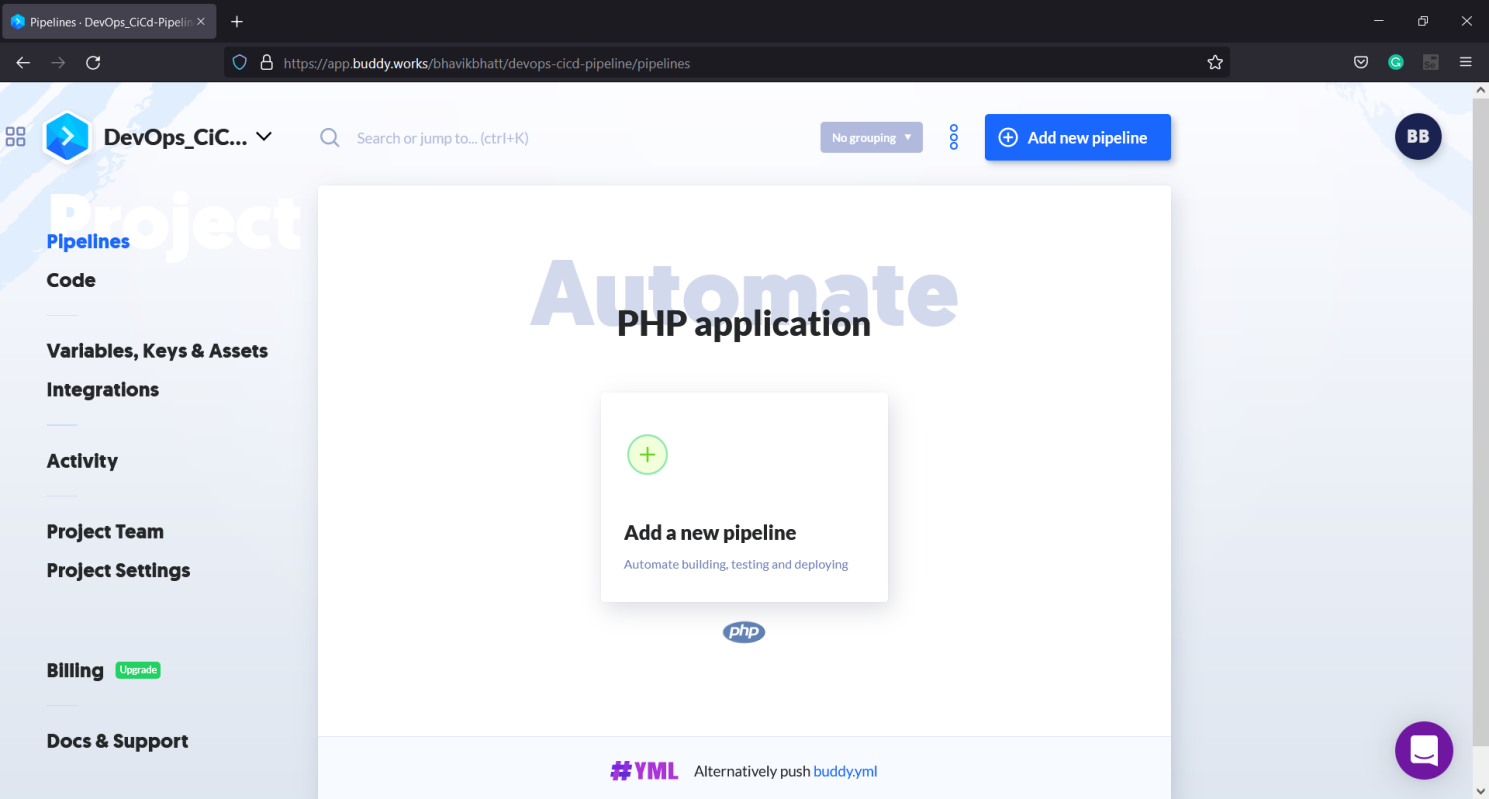


1. **PIPELINE CREATION**

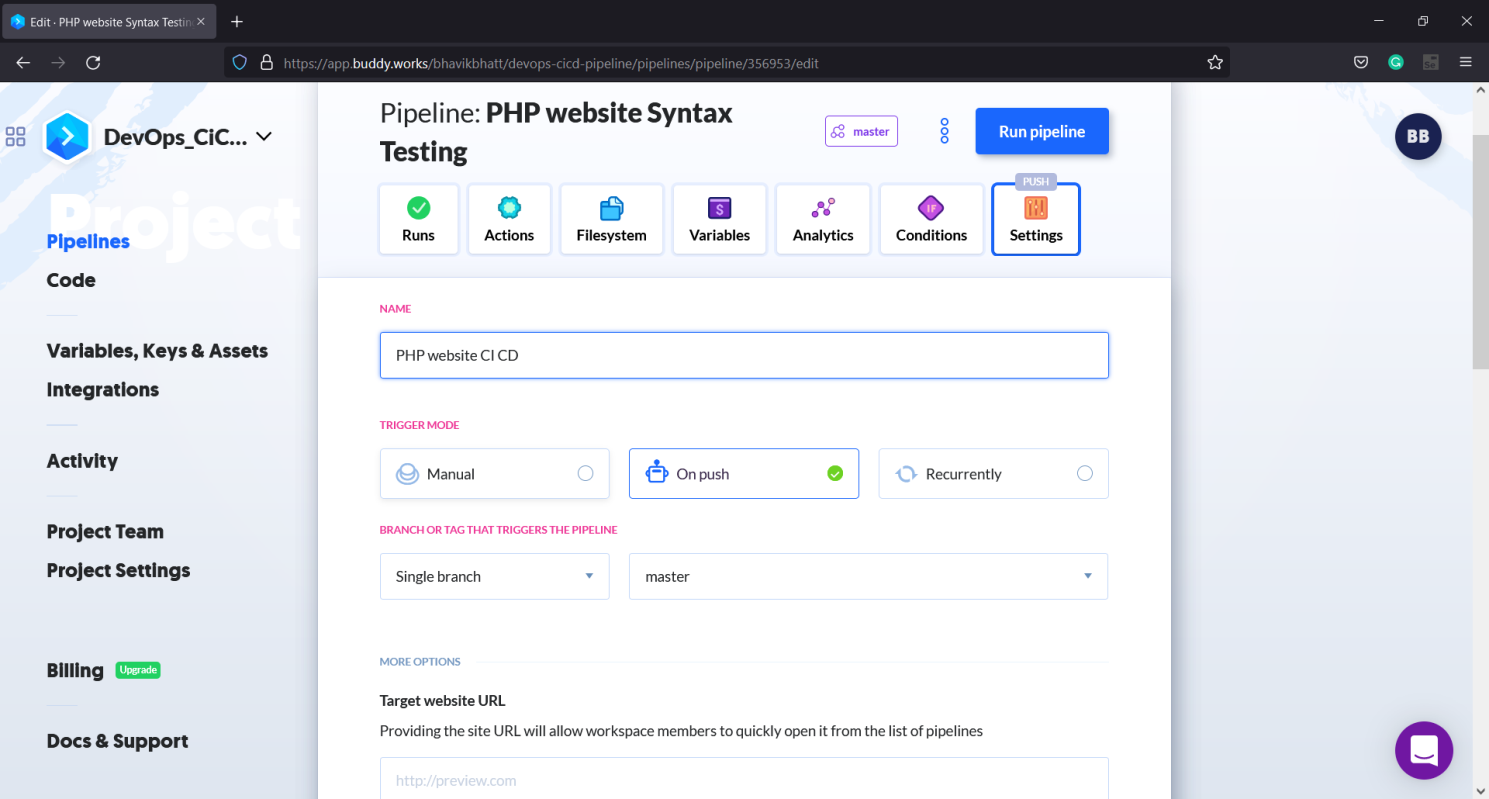
* Creating a new project and linking it with GitHub repository



* Project Created



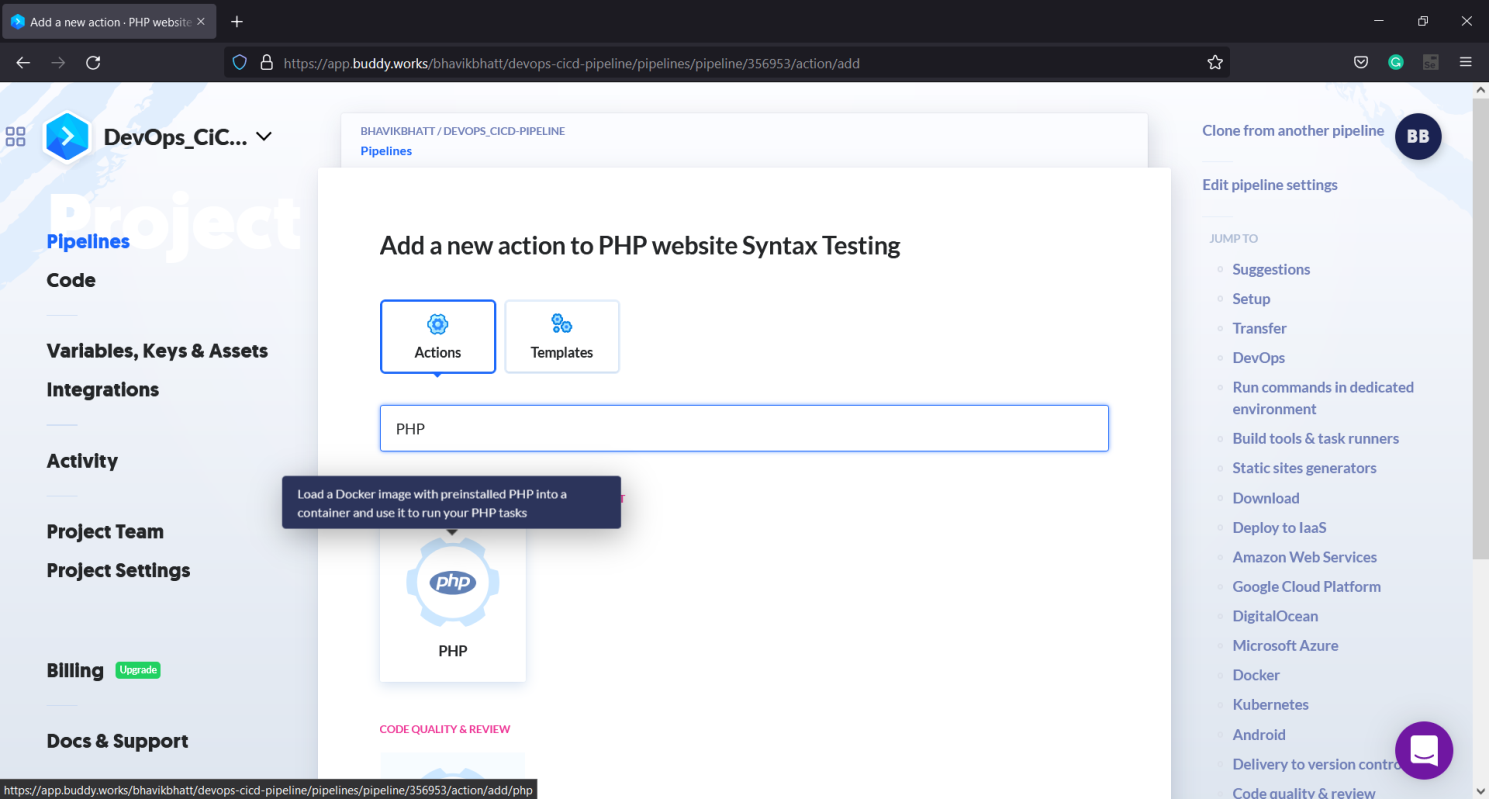
* Adding a pipeline in the project



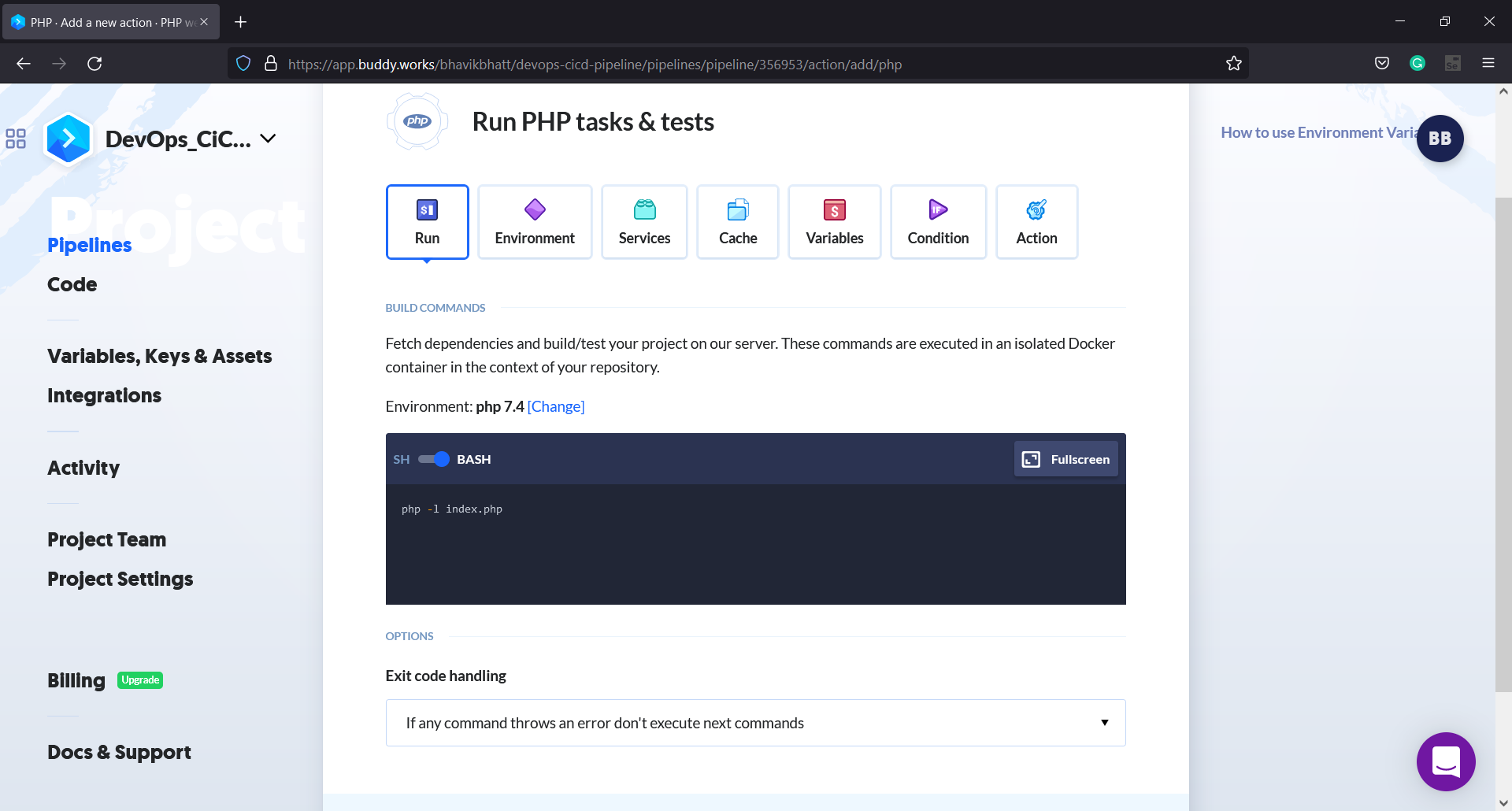
Setting the trigger mode as “on push” to execute the pipeline after every commit on GitHub repository.

1. PRIMARY ACTION 1: PHP TESTING

* Selecting PHP action

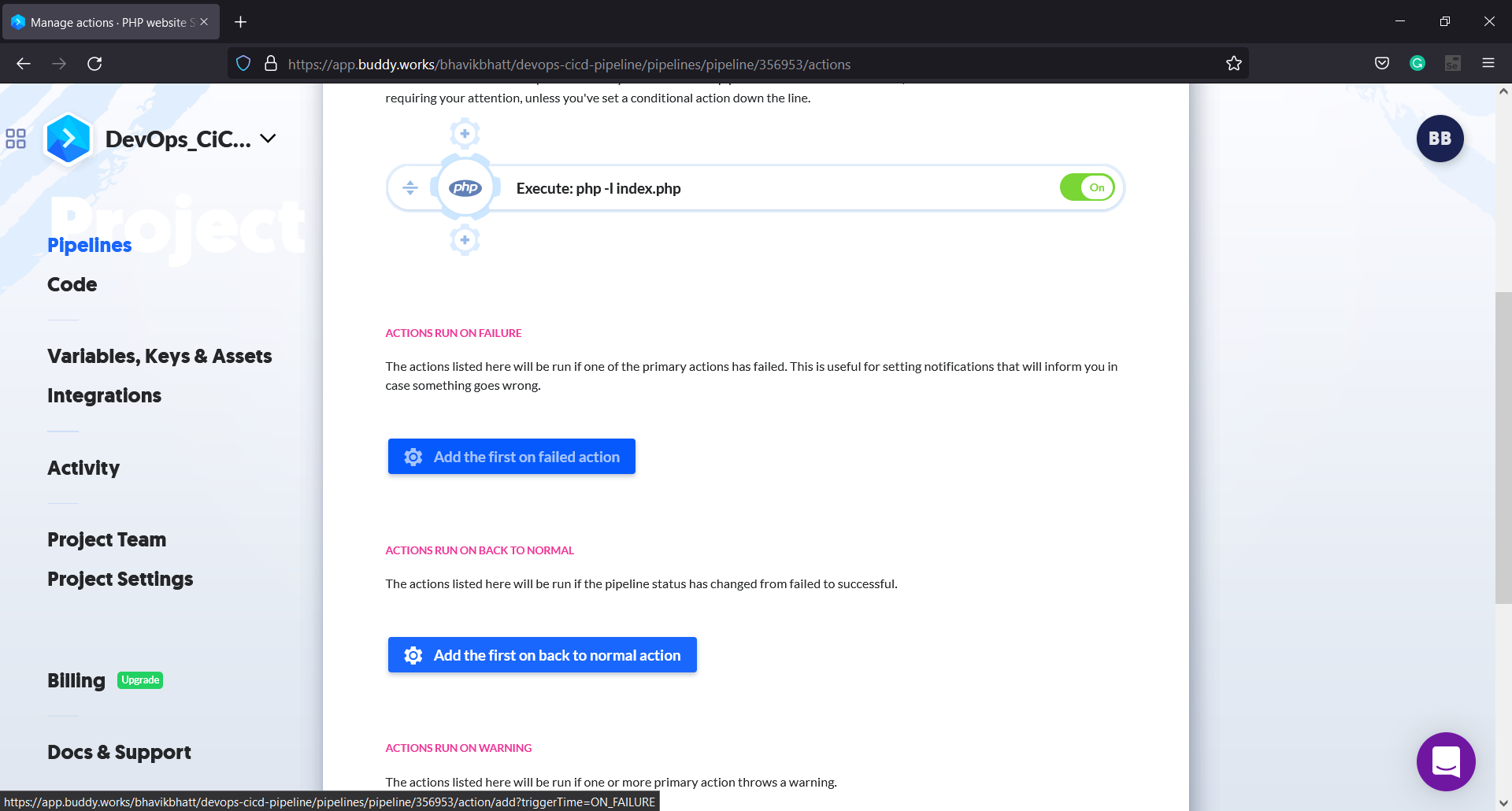


* Adding the test command

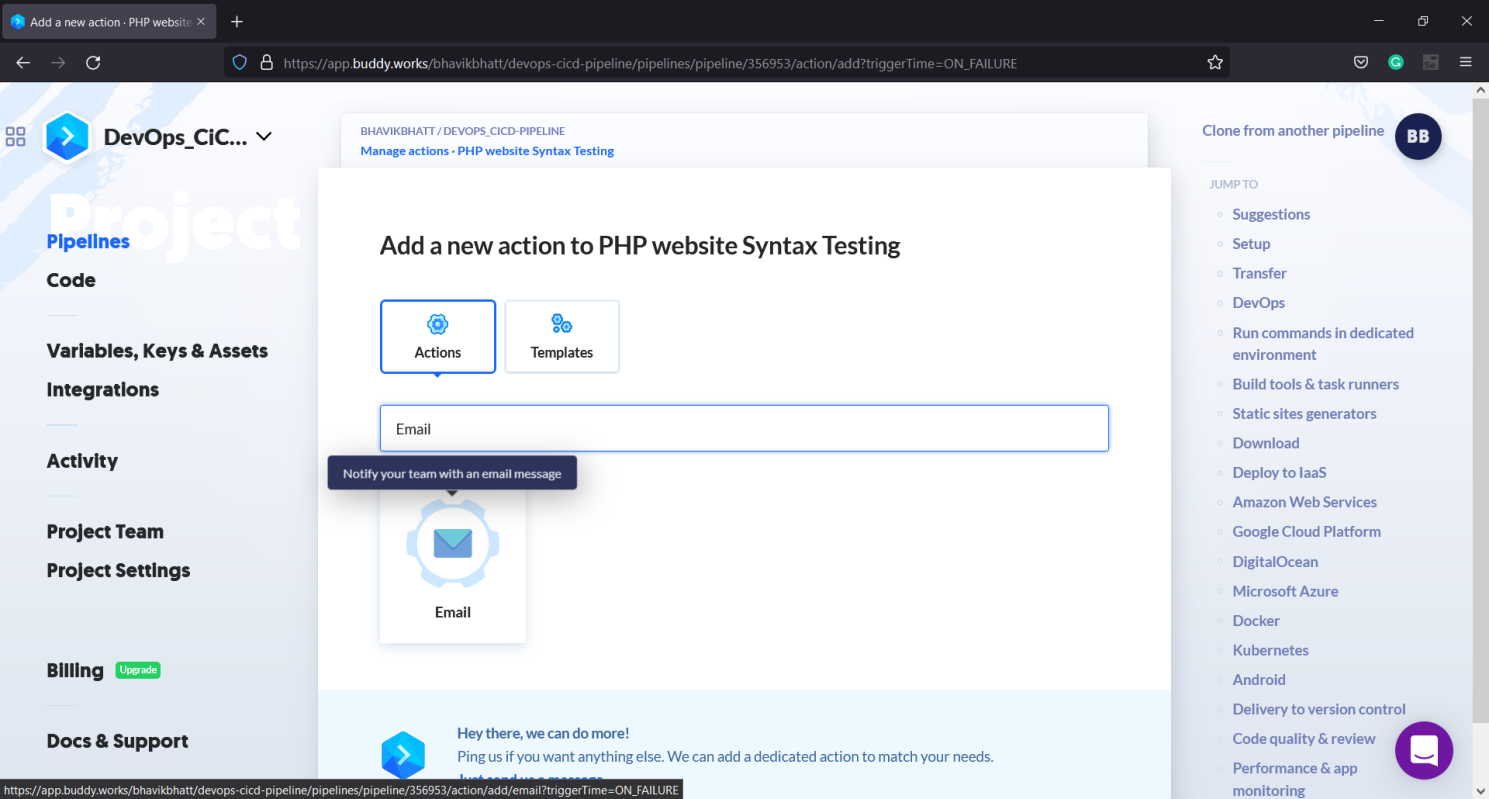


1. PRIMARY ACTION 2: GULP
2. ACTION TO RUN ON FAILURE: EMAIL NOTIFICATION

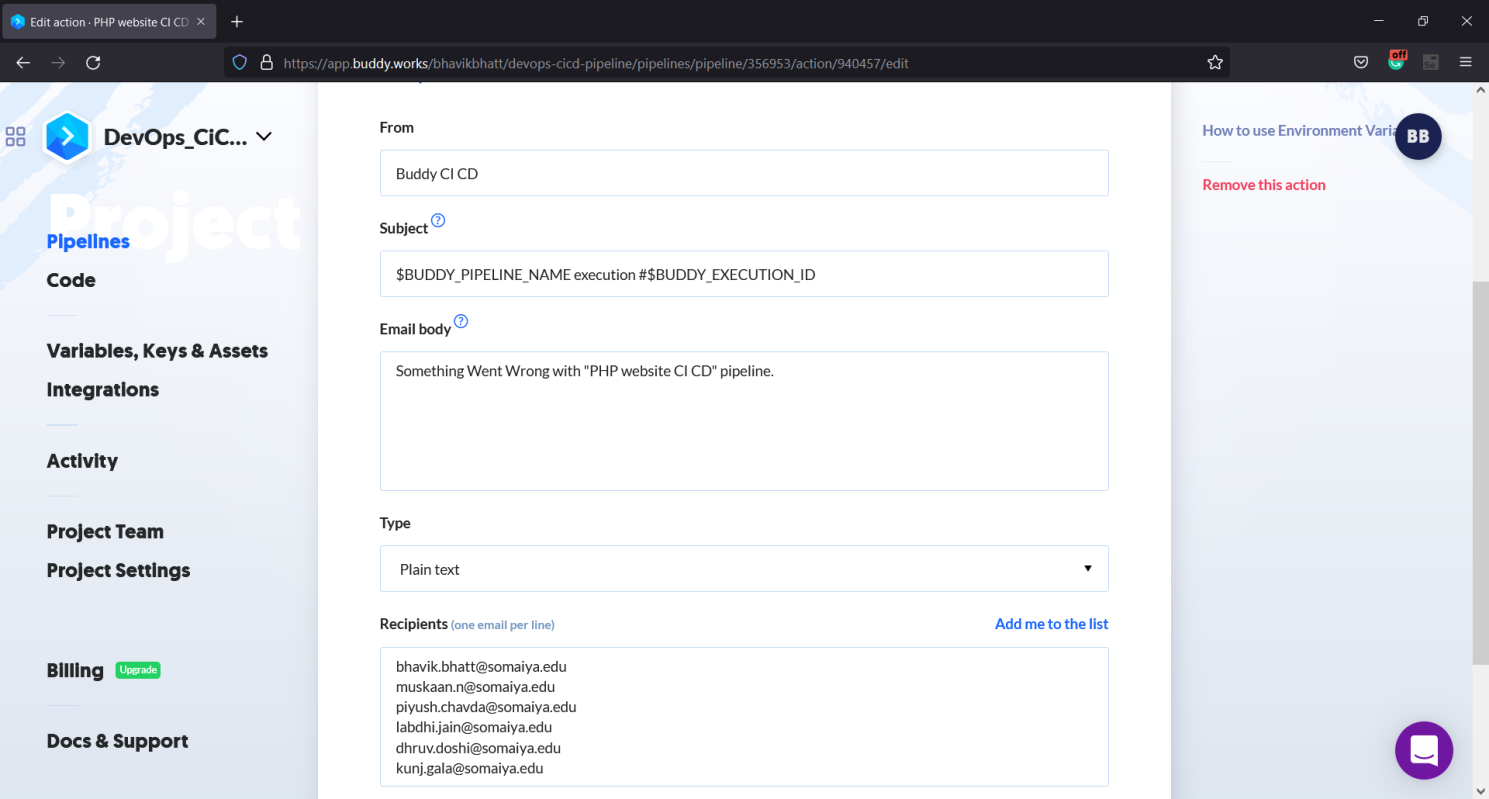
* Configuring action to run on failure



* Selecting Email notification



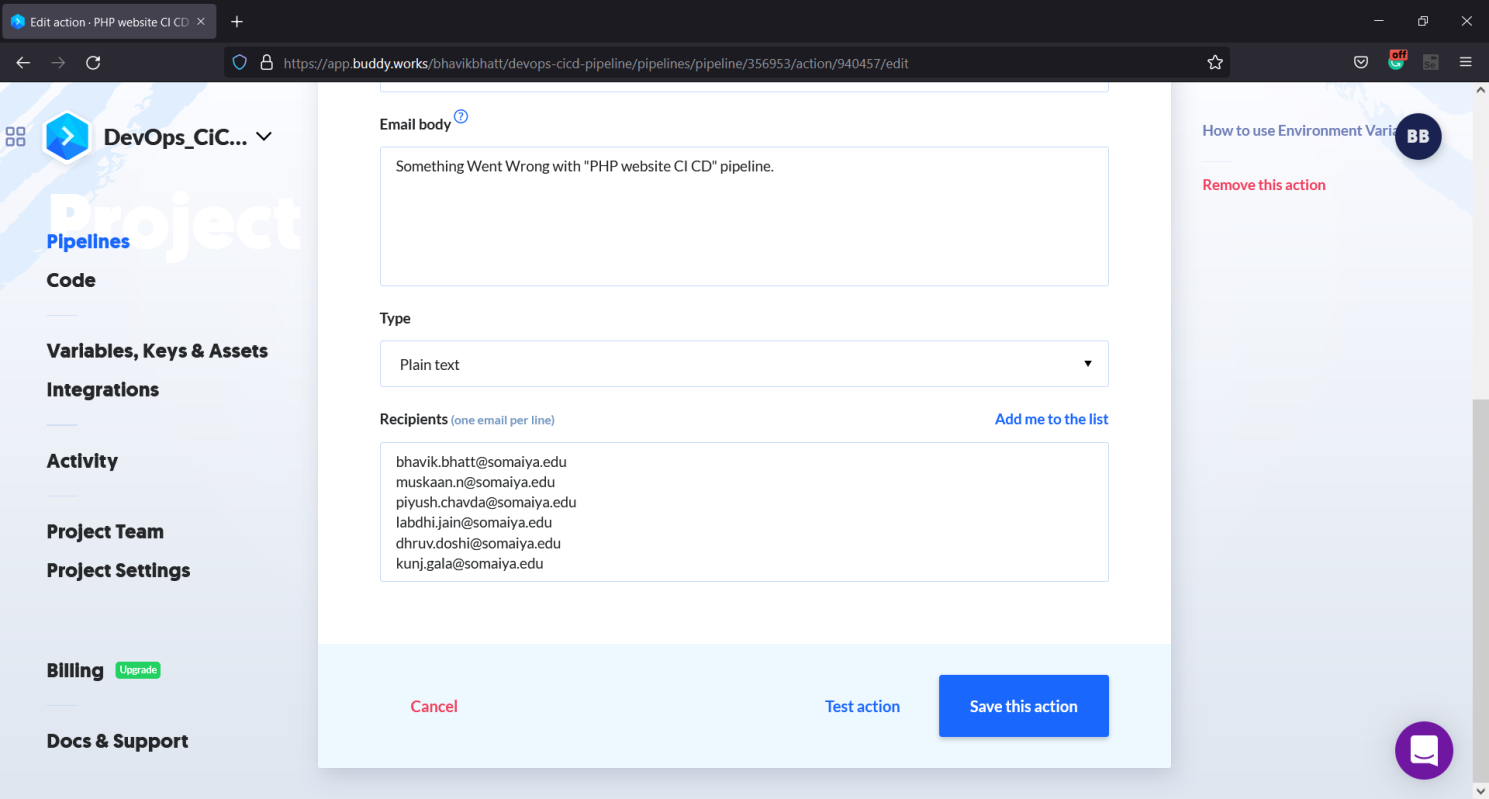
* Setting up the Email notification

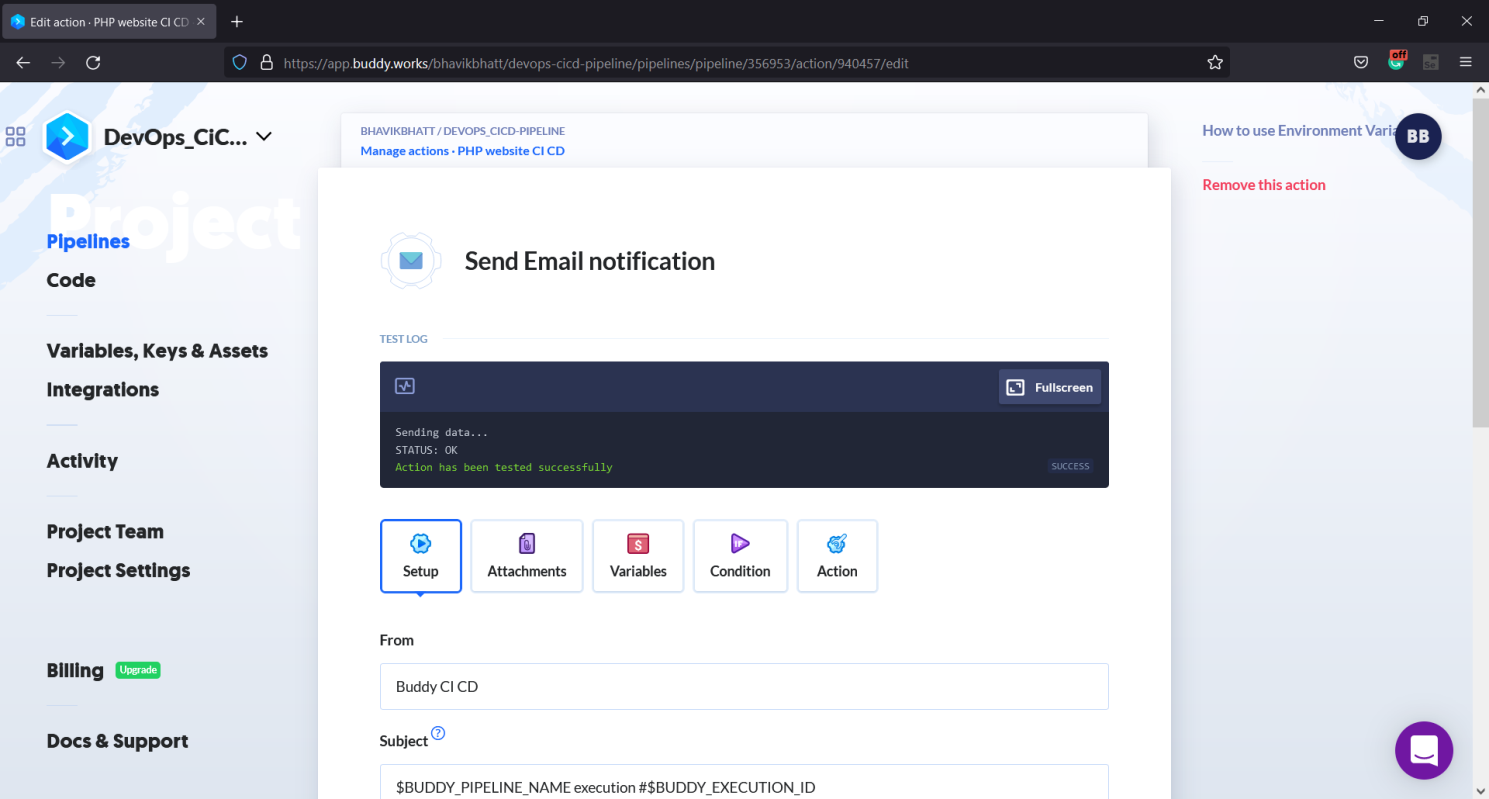


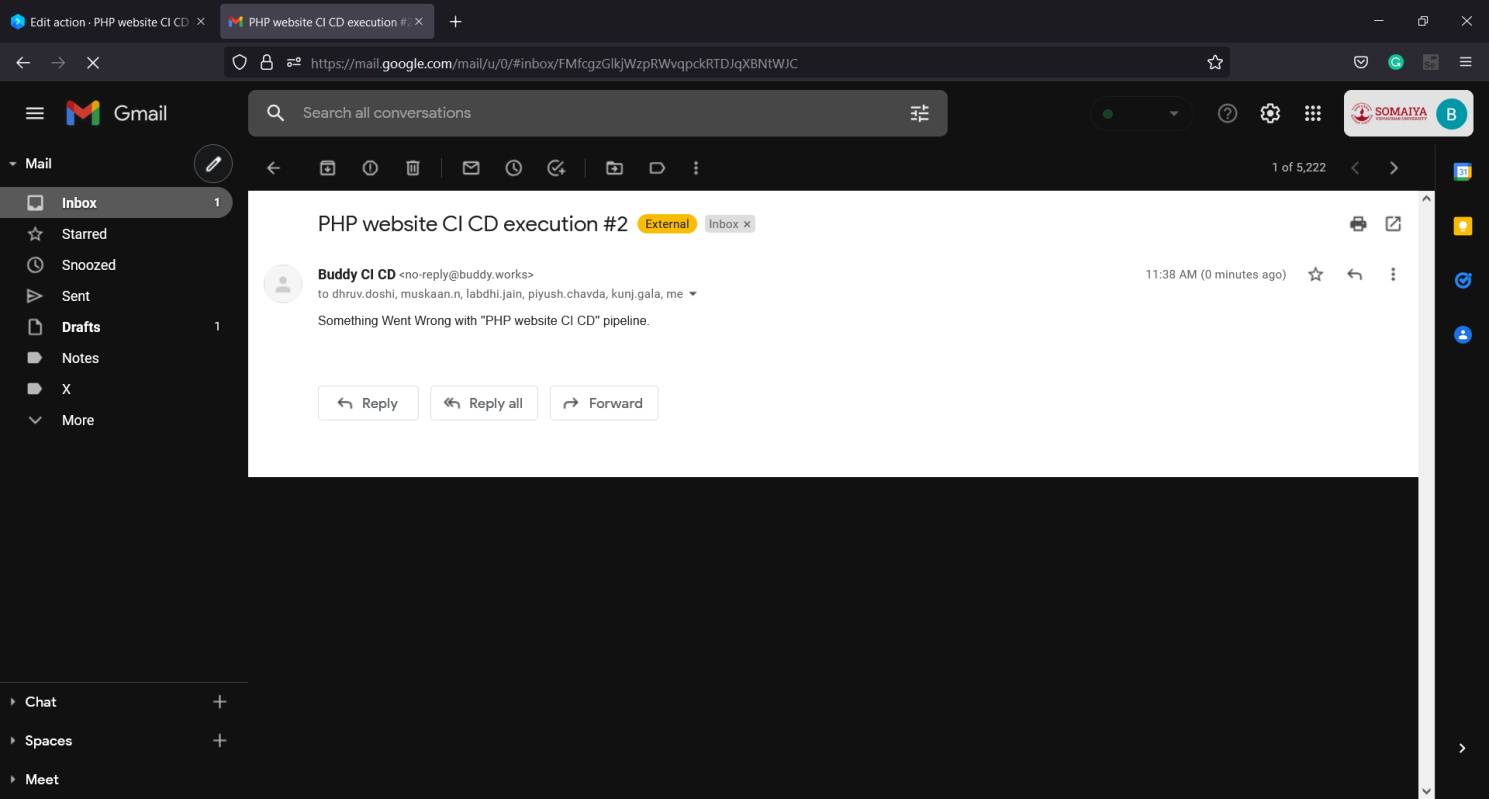
Customizing email body and Subject as per needed.

Recipients includes all the members of the teams.

* Testing the action







1. **PIPELINE EXECUTION**
2. **CONCLUSION**

In this project, we have created a simple PHP application and have built a CI/CD pipeline using Buddy which is self-hosted continuous integration and delivery softwareand have used git and github for source code management and version control, PHP command lines for testing of the application and Github Pages for

deployment of the website. Whenever there is a new commit on Github, this pipeline will fetch the code, install the necessary dependencies, build and test the application again and deploythe changes on github pages seamlessly