

Portfolio Website (Landing Pages)



Objectives

The main objective of building this website is to demonstrate the idea of landing pages as per the problem statement.

Also, the portfolio website can be utilized as a demo template to for users that wish to design their own website based on front end development only.

Benefits

- Easy to demonstrate landing page concept.
- Based on front end development only.
- Can be used as a demo template.
- Editable and can be updated as per user's convenience.

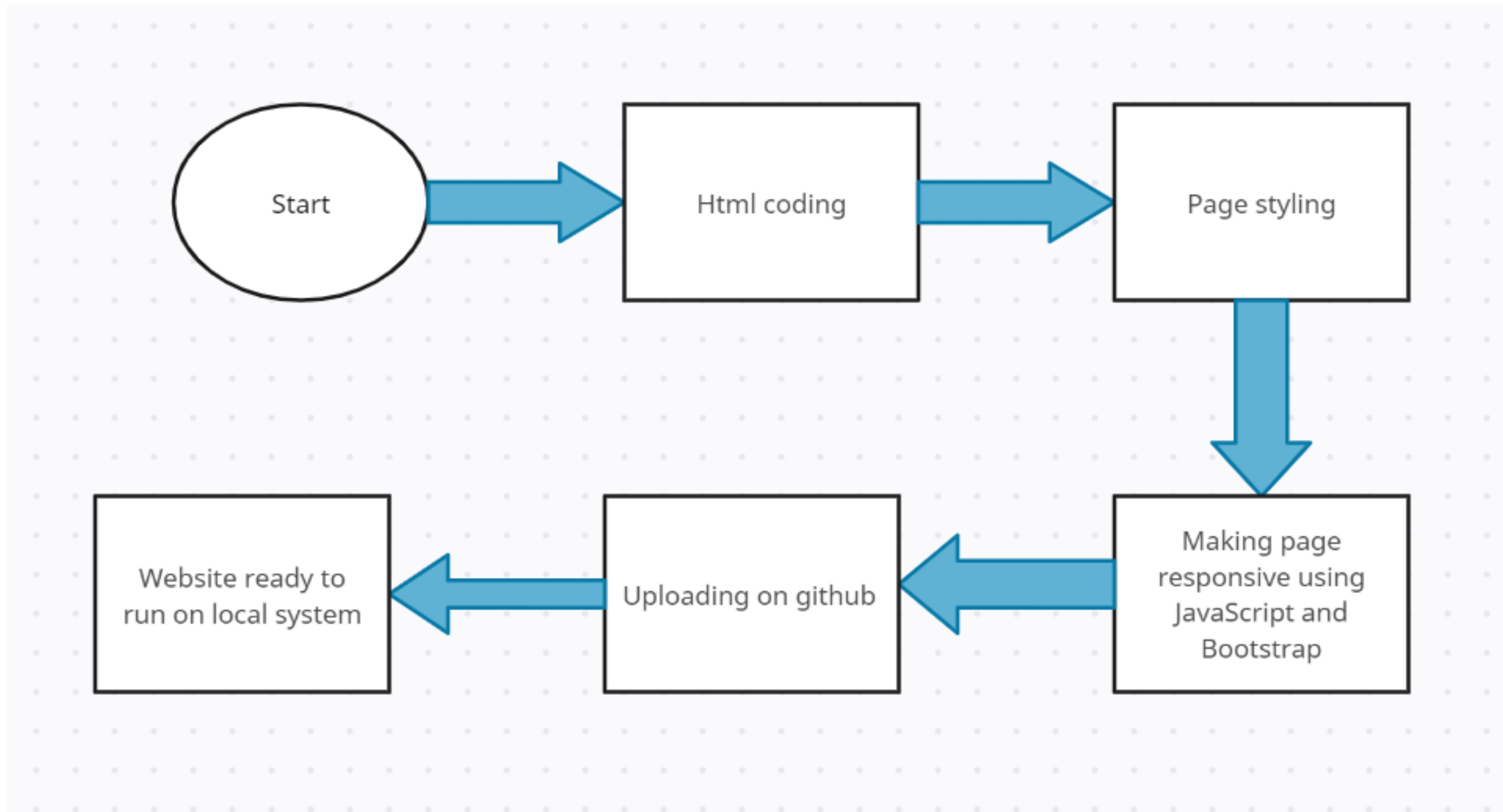
Data Sharing Agreement

This website requires the owner's data that too during development phase which are:

- **Name**
- **Contact Information**
- **Social Media Links**
- **Skills**
- **Domains of Interests**
- **Achievements**
- **Experiences**
- **Academic details**

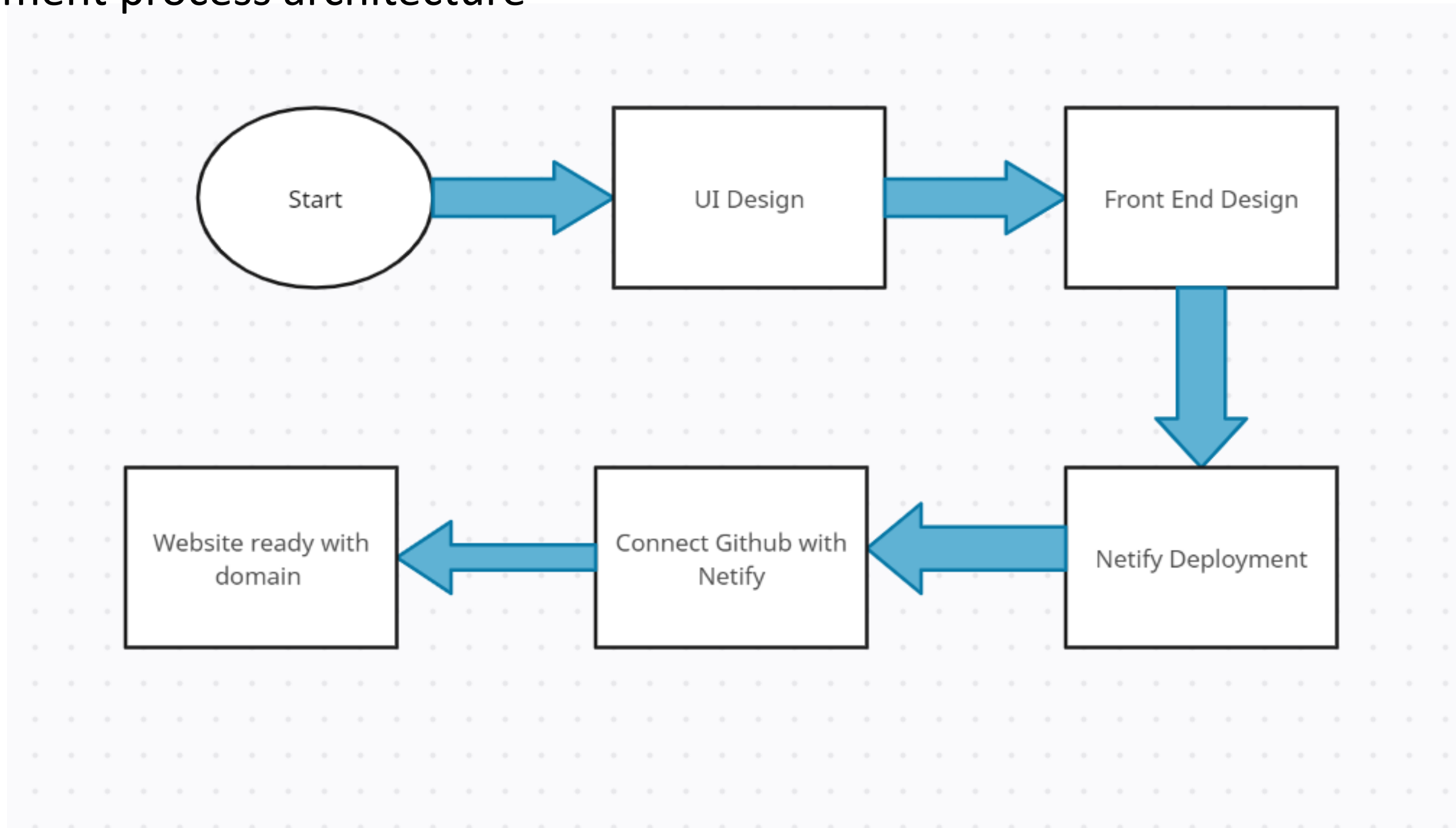
If owner wishes to keep the information limited, then it is also possible

Architecture



Architecture

Deployment process architecture



Step 1: UI/UX Design

- I used Figma to design the Ui for the homepage.
- First I chose the drawboard and put on some shapes to divide the header part.
- Then detailed the area with the menu icons on the header
- After that I chose the area for name and description.
- In the similar way I designed the other pages from the menu.

Step 2: Gathering Images

- I used favicon.io to download favicons and wallpaper cave to download the background.
- For the social media icons I used favicons as well.

Step 2: Linear and Lasso Regression

- We first tried to create a base Linear regression model to identify the relation with different parameters.
- With Linear regression, we found, PM2.5 is highly correlated with AQI.
- Every parameter we took was sparse and therefore haven't shown any kind of relation with other parameters.
- Initially we got 83.97% R^2 for training and 85.6% R^2 for testing dataset, which shows overfitting. But after visualization, it was clear that Linear regression is not a right model.
- After Cross-Validation, we got the high MSE.
- Even after implementing Lasso Regression, we got the similar result. Thus we moved on to another model.

Step 3: Creating and styling pages

- Using Html and CSS I designed and styled the webpage and used bootstrap documentation for further division of data within the page.
- Using JavaScript I made the pages responsive by invoking functions and loops for switching to different menu and stylings during interacting with images.
- I also imported various documentation from mdn docs for smooth transition of website.

Step 4: Sorting files and Uploading

- In the atom editor I sorted the files into folders for images css, js, vendors etc.
- After that while navigating to GitHub account I created a new repository where I uploaded the files in the same sequence and sort.

Step 5: Deployment

- For the deployment of the website I used the netify website.
- First of all I linked the repository to the netify website then chose the domain name for the website.
- After agreeing to certain T&Cs and waiting for few seconds the website was deployed and ready to be used.
- Url : [Portfolio \(portfoliovieh.netlify.app\)](https://portfoliovieh.netlify.app)

Frequent Q&A

Q) What is the complete flow of your project?

- Refer to slide no 4 and 5 for better understanding.

Q) How did you choose the project?

- Looking to the technical trends and convenience of development I chose to build this website as with it is easy to understand the problem statement.

Q) What are the different stages of deployment?

- When the files were ready and developed, they were uploaded on GitHub.
- Then on netlify website I linked the GitHub repository where these files were uploaded.
- Then chose the domain name for the website link.
- Chose the owner's name and did some deployment settings.
- And after few moments of waiting, our website was ready and deployed.
- The deployed website can be found with url: [Portfolio \(portfoliovieh.netlify.app\)](https://portfoliovieh.netlify.app)