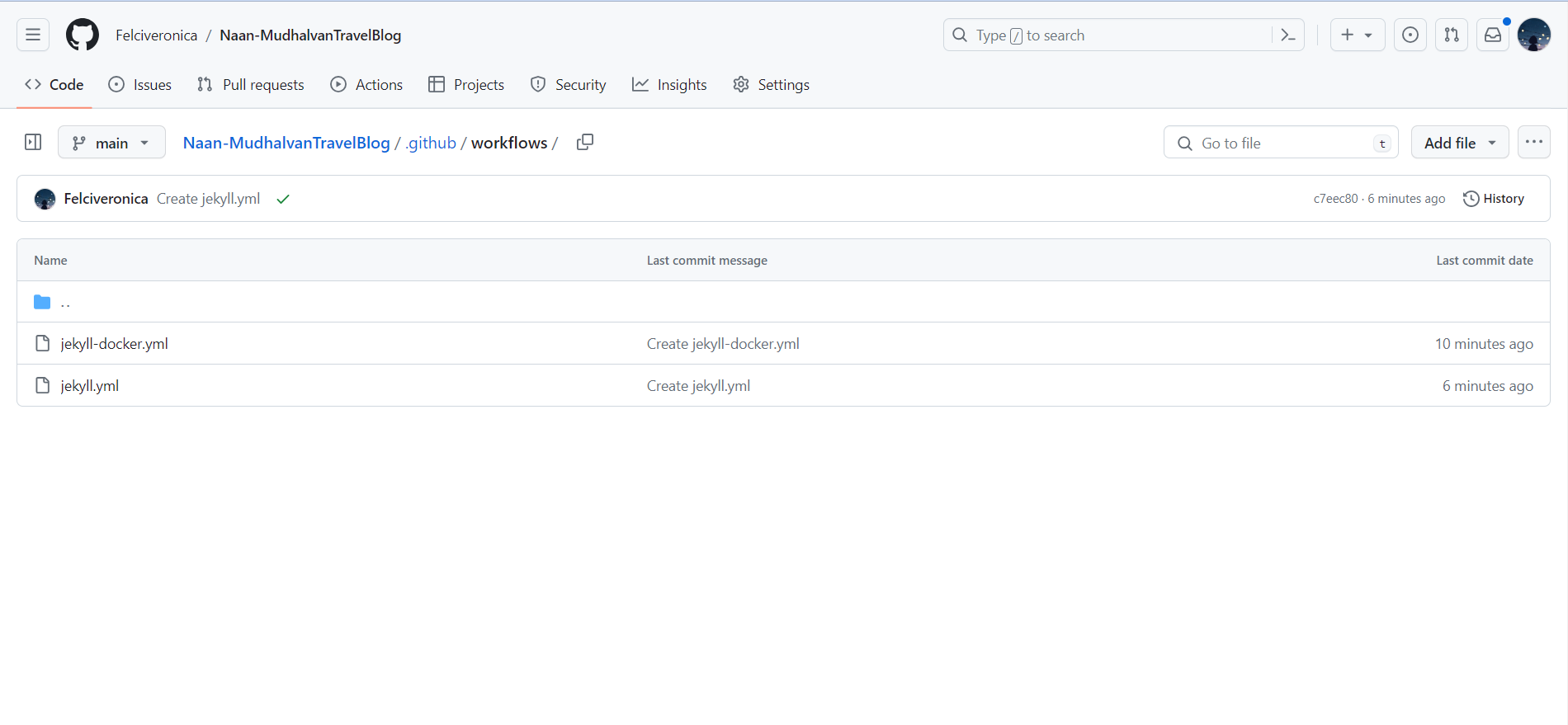
**Phase 4:**



**Procedure:**

These steps include checking out the code, building, testing, and deploying your application. Customize the commands according to your project's requirements.

* **Define Workflow Steps:**

After Select a CI/CD service that suits our project we create a configuration file that defines our CI/CD pipeline. The format and details of this file will depend on the CI/CD service we choose. In GitHub Actions, you'd create a YAML file (e.g., .github/workflows/jekyll.yml).

* **Set Trigger Conditions:**

Define when the pipeline should be triggered. Common triggers include code pushes, pull requests, or tags. Specify the branches or events that should initiate the pipeline.

* **Set Trigger Conditions:**

Specify the deployment targets, such as cloud platforms or servers, and provide authentication details.

* **Push the Configuration File:**

Commit and push the CI/CD configuration file to your Git repository.

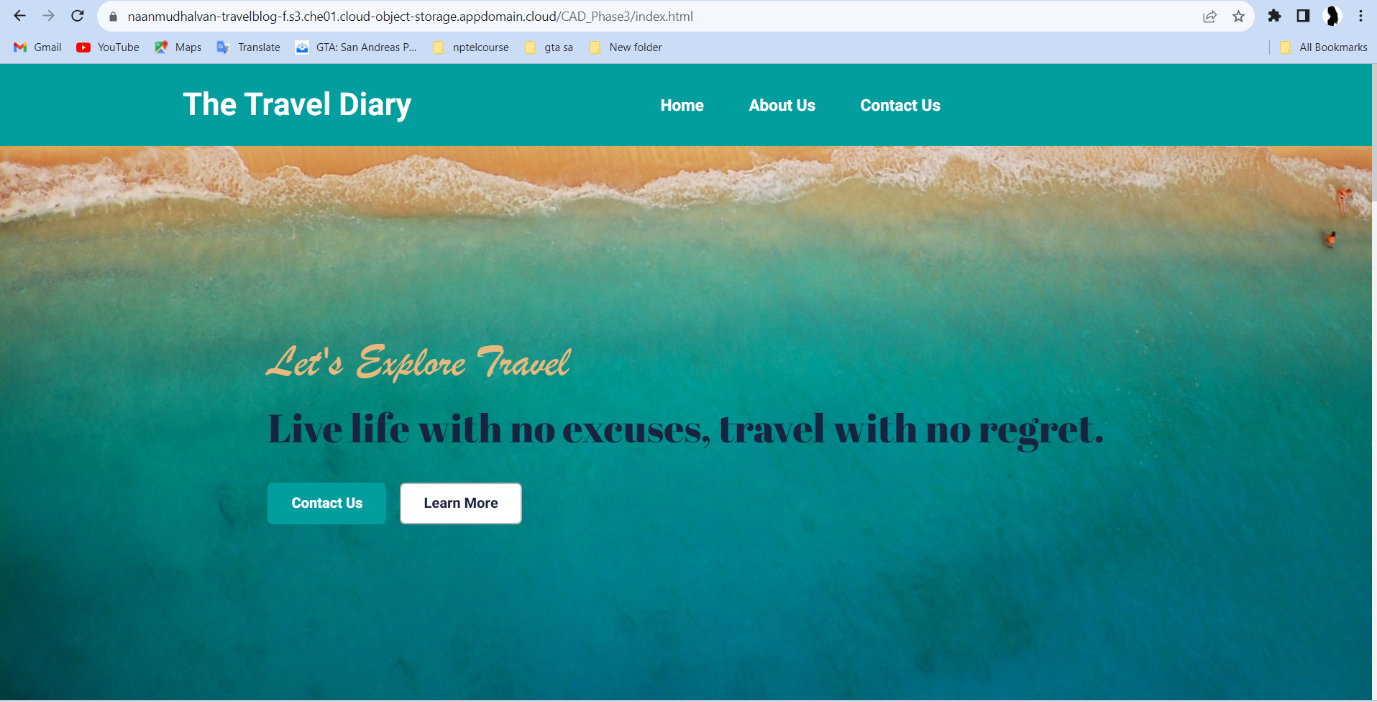
Configuring the service to recognize the YAML configuration file you added to your repository. The CI/CD service will automatically run the pipeline when the trigger conditions are met. It will check out the source code, build the website, and deploy it to IBM Cloud.

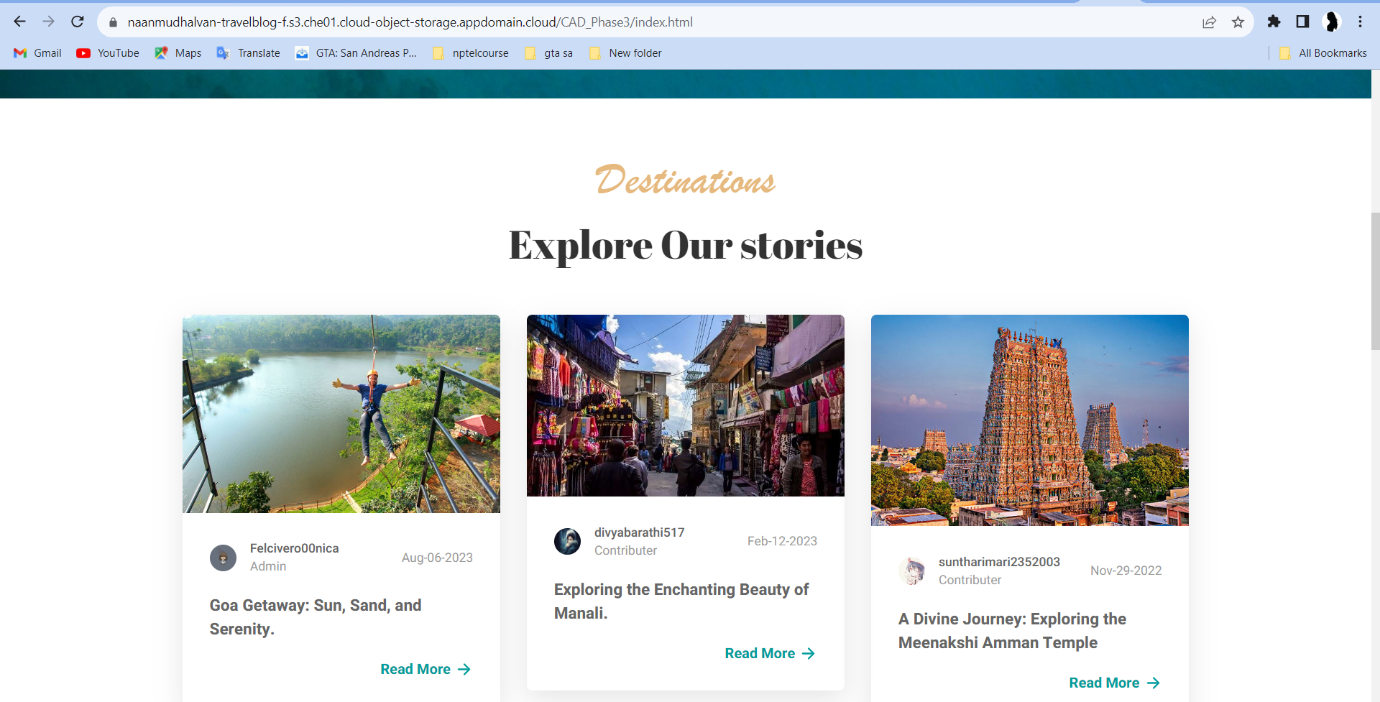
After Configuring the .yml file the html files and css files are uploaded and their acces is made public.

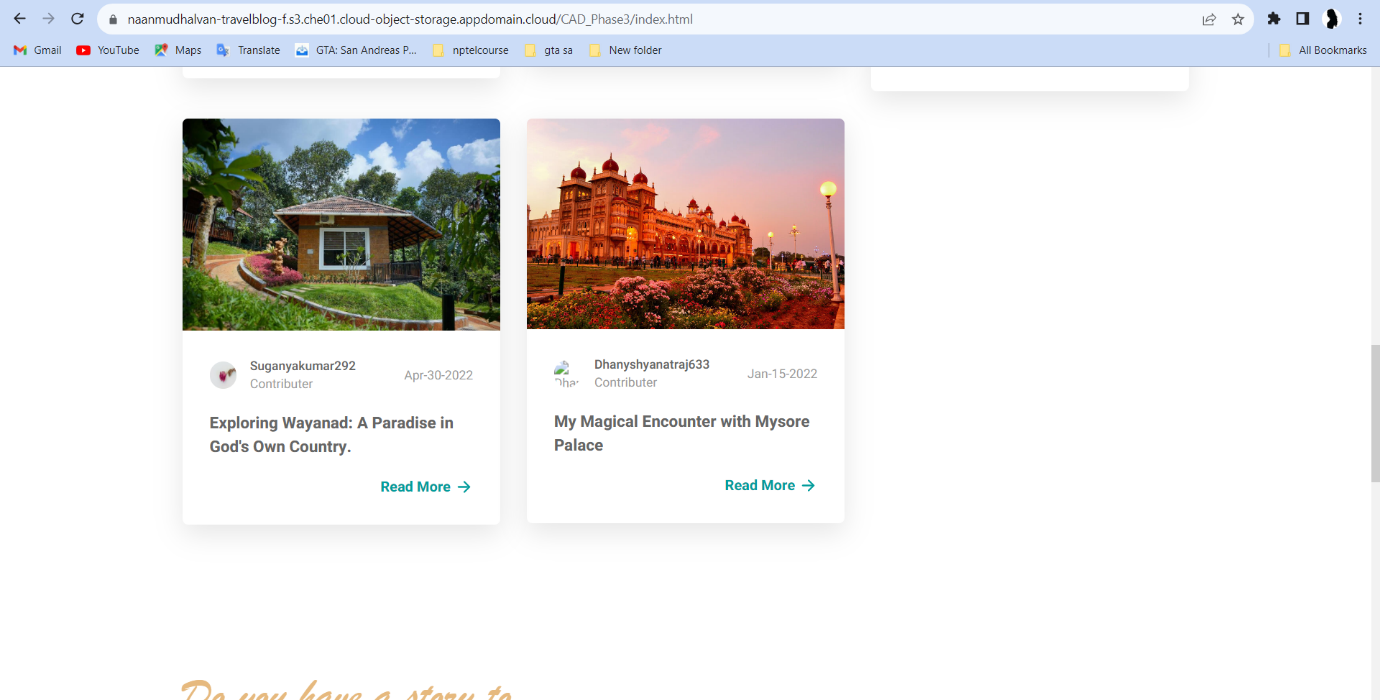
Static blog link: <https://travel-blog-divya.s3.che01.cloud-object-storage.appdomain.cloud/CAD_Phase3/index.html>

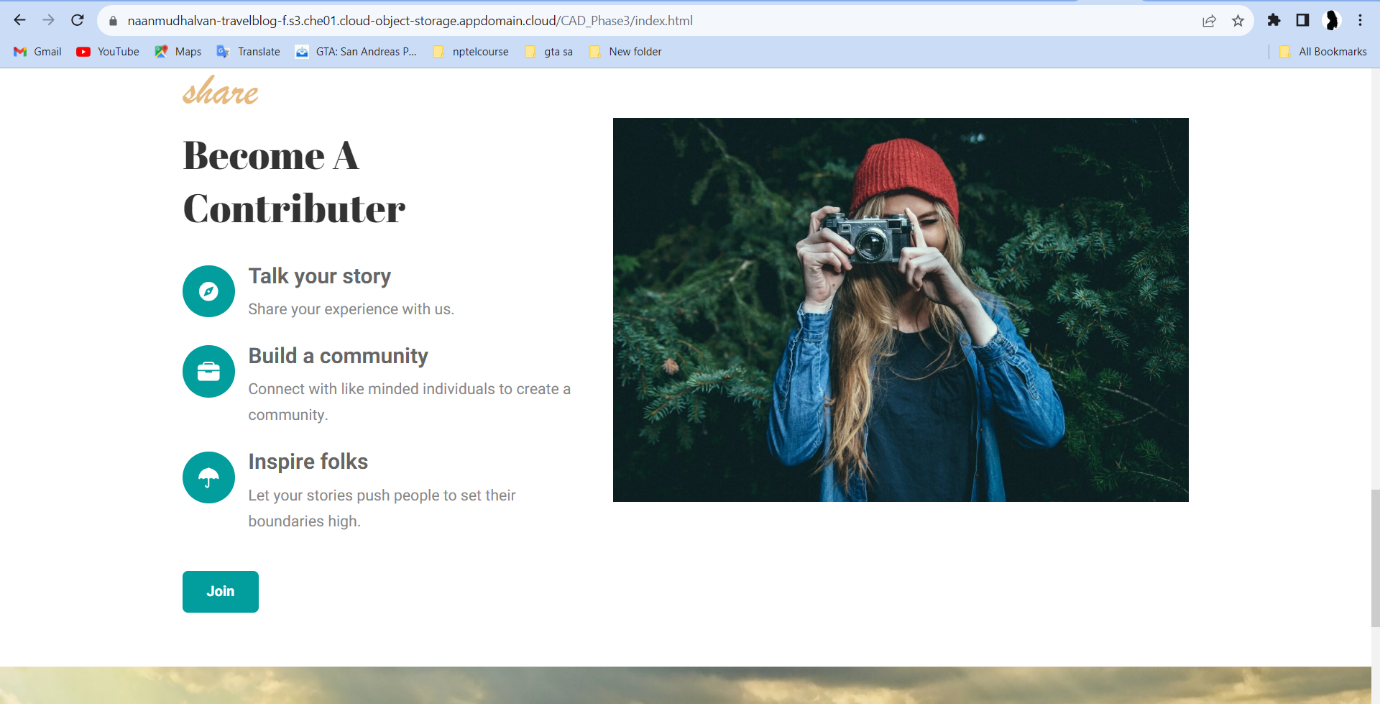
Jekyll.yml link:[/.github/workflows/jekyll.yml](https://github.com/Felciveronica/Naan-MudhalvanTravelBlog/blob/c7eec8055af2eae49823b7edef47ba78546a1fcf/.github/workflows/jekyll.yml)

Jekyll-docker.yml link:[/.github/workflows/jekyll-docker.yml](https://github.com/Felciveronica/Naan-MudhalvanTravelBlog/blob/c7eec8055af2eae49823b7edef47ba78546a1fcf/.github/workflows/jekyll-docker.yml)







[](https://github.com/Felciveronica/Naan-MudhalvanTravelBlog/blob/c7eec8055af2eae49823b7edef47ba78546a1fcf/.github/workflows/jekyll-docker.yml)