PORTFOLIO WEBSITE BY UCHE CHINEDU

November, 2022

Module Code: B9IS109

MODULE LEADER: OBINNA IZIMA

STUDENT NUMBER: 10594644

TABLE OF CONTENTS

References

LIST OF FIGURES

1 Introduction

In today's technology-driven world, the idea of a paper portfolio is on the verge of extinction and everyone wants to make their existence known online. Clients can easily get in touch with site administrators through online platforms such as the Portfolio website. Django is a free and open source Python-based he web framework that follows the Model-Template-Views (MTV) architectural pattern (Django, 20220; Holovaty, Kaplan-Moss, 2013). Django's main goal is to simplify the creation of complex database-driven he websites. The framework emphasizes the principles of component reusability and "pluggability", less code, low coupling, rapid development, and non-repetition. Python is used throughout, including configuration, files, and data models. Django also provides an optional create, read, update, and delete management interface. It is dynamically generated by introspection and configured by the administrative model. The Django portfolio website is an interactive portfolio website that gives users the opportunity to interact directly with the owner. Portfolio sites give as many people as possible the opportunity to read about you and others, apart from being contacted by clients. While there are no mandatory filters or bundled apps to run a Django project, reusable apps typically rely on them, so developers should continue to use the official stacks to help the ecosystem of apps can be fully utilized (Baumgartner and Malet, 2015),

2 Research and Planning

The Portfolio website has been created for users who wish to contact us about some of our services. Many may offer the same services as we do, but having access to all the information from your portfolio website makes life easier for our clients and potential clients. Initially, we

use Django Python as our framework. Bootstrap is used to create web applications that work on both desktops and mobile devices. The backend of this web application uses Sqlite3.

3 Choice of Framework and Technologies

3.1 Python

Python version 3.10.6 has been implemented in this web application. Python is used to create web applications on the server. This web application was developed exclusively in Python Django. Python is used throughout the project. The web application's settings, files, and data model are developed using Python. Web applications are developed in virtual environments.

3.2 Django framework

Django is a free, Python-based, open-source web framework that helps developers easily build complex, database-driven web applications. The Django framework encourages code reusability, rapid development, and clean, practical design for developers. Django supports his five database backends: PostgreSQL, MySQL, Oracle, Microsoft SQL and MariaDB. All installed applications, external libraries, web application backend settings, debug settings, etc. are developed in the main project file in the settings.py file.

Being a web template, Django needs a convenient way to generate HTML dynamically. The most common approach relies on templates. A template contains the static part of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.

A Django project can be configured with one or several template engines (or even zero if you don’t use templates). Django ships built-in backends for its own template system, creatively called Django template language (DTL). Backends for other template languages may be

available from third-parties. Django defines a standard API for loading and rendering templates regardless of the backend. Loading consists of finding the template for a given identifier and preprocessing it, usually compiling it to an in-memory representation. Rendering mean interpolating the template with context data and returning the resulting string. The Django template language is Django's own template system. It's a good template library even though it's fairly opinionated and sports a few idiosyncrasies. If you don't have a pressing reason to choose another backend, you should use DTL, especially if you are writing a pluggable application and you intend to distribute templates. Django's contrib apps that include, like Django.contrib.admin, use DTL.

3.3 SQLite

SQLite is a database engine. Software that allows users to interact with relational databases. In SQLite the database is stored in a single file. This is a feature that sets it apart from other database engines. This fact allows for a high degree of accessibility. Copying a database is less complicated than copying the files in which the data is stored. Sharing a database can mean sending email attachments. SQLite does not reject values of the wrong type. You could accidentally put the wrong data type in the column. Storing different data types in the same column is a bad practice that can lead to errors that are difficult to fix. So it's important to use schema safely, even if SQLite doesn't enforce it. The benefits of accessing and manipulating databases without using a server application are immense. SQLite is used all over the world for testing, development, and other scenarios where it makes sense to have the database on the same disk as the application code. The maintainers of SQLite consider it one of the most replicated software in the world.

3.4 Bootstrap Template

Bootstrap is a dynamic front-end framework for developing modern websites and web applications. There are a large number of HTML and CSS templates for UI elements such as buttons and forms, and they are free to use. Bootstrap also supports JavaScript extensions. Bootstrap templates are pre-designed so users don't have to worry about creating a set of themes. With Bootstrap, templates are attractive, modern, and most importantly easy to use. The bootstrap template requires the user to make changes as desired in the web application and discard the changes that the template entails.

3.5 Heroku

Heroku is a cloud platform as a service (PaaS) that runs on containers (PaaS). Heroku is the platform developers use to deploy, manage, and scale modern programs. Our platform is attractive, customizable and easy to use, making it the fastest way for developers to bring their products to market.

4. User Experience Design

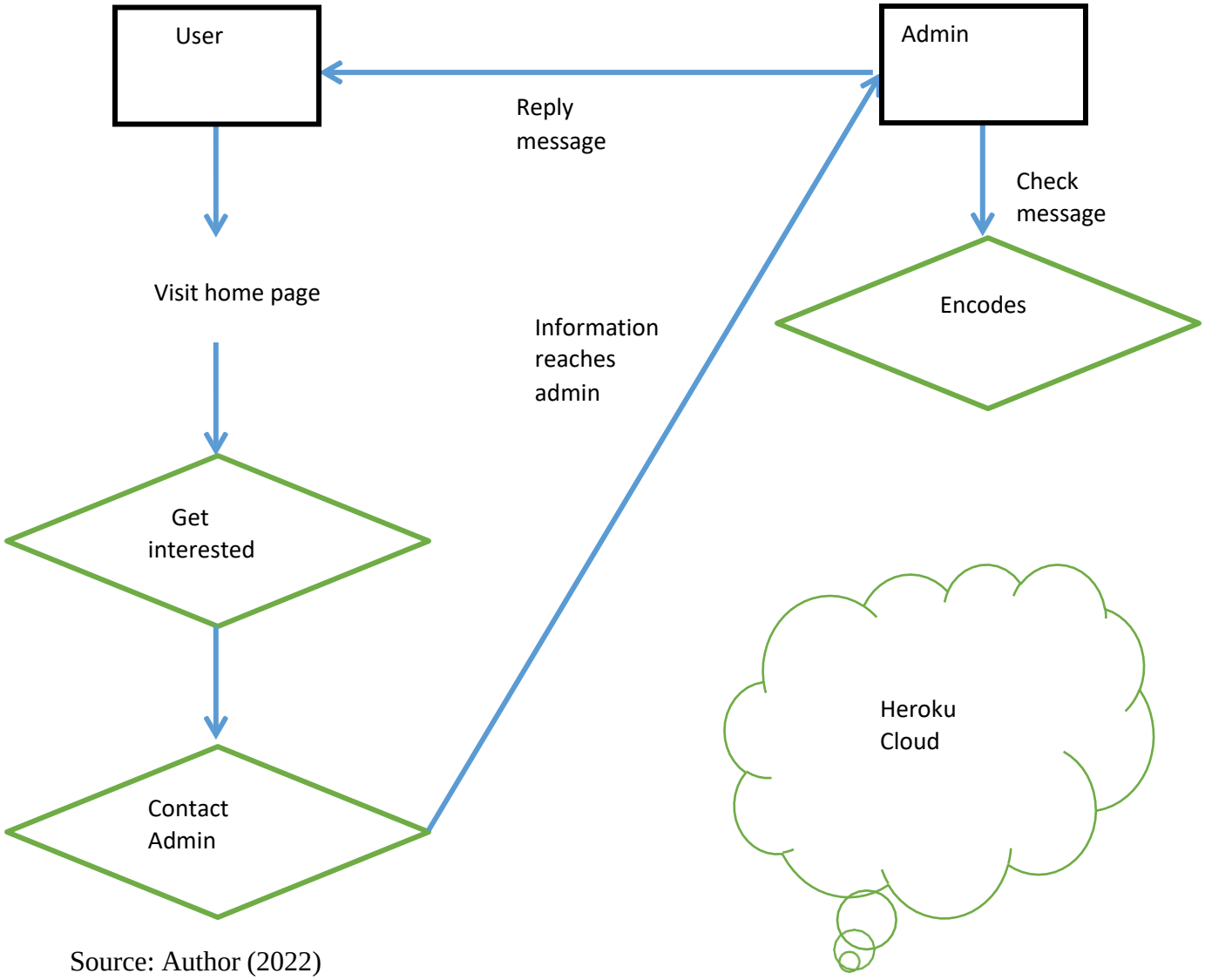
User Experience (UX) Design is the process that design teams use to create products that provide meaningful and relevant experiences for their users. UX design involves designing the entire process of product procurement and integration, including aspects of branding, design, usability, and functionality.Experience design includes not only software usability, but also , including designing other experiences related to the product. B. Marketing Campaigns, Packaging, and After-Sales Service. Most importantly, UX design is about providing solutions that address problems and needs. After all, no one uses a product that serves no purpose. The UX of these

websites includes Python, Django admin, and Heroku cloud deployments to improve user satisfaction. A flowchart of the design activities is provided in the next section.

4.1 Flow Diagram

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a schematic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as different kinds of boxes and connects the boxes with arrows to indicate their order. This schematic shows a solution model for a particular problem. Flowcharts are used to analyze, design, document, or manage processes or programs in various areas. Flowcharts show the flow of information through users and administrators. The user accesses the site as shown. Via the home page, query the site owner's skills and determine if they are using the service. He looks at the skills and portfolios of site owners and is interested in what he sees. If you're not interested, the sequence ends, but if you're interested in contacting the site owner, we'll send a message to the admins. The administrator then receives the message, examines it, and decrypts it. If the requested service can be performed, the administrator sends the user feedback her message. The flow chart is shown in Figure 1 below.

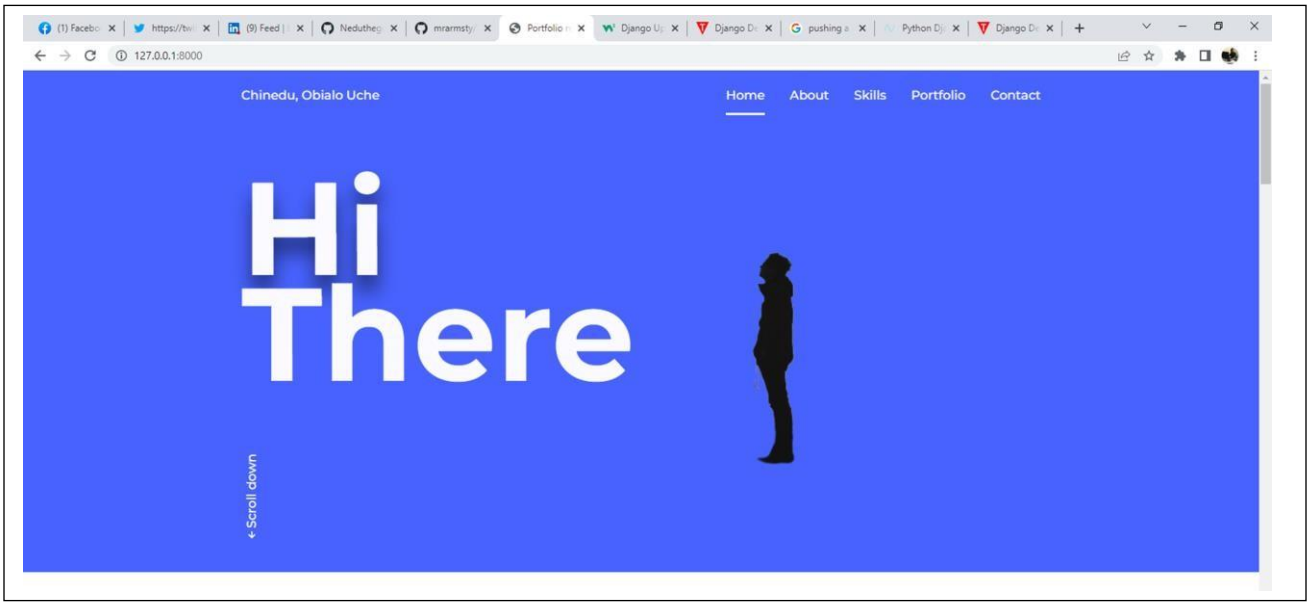
Figure 1: Flow diagram



Source: Author (2022)

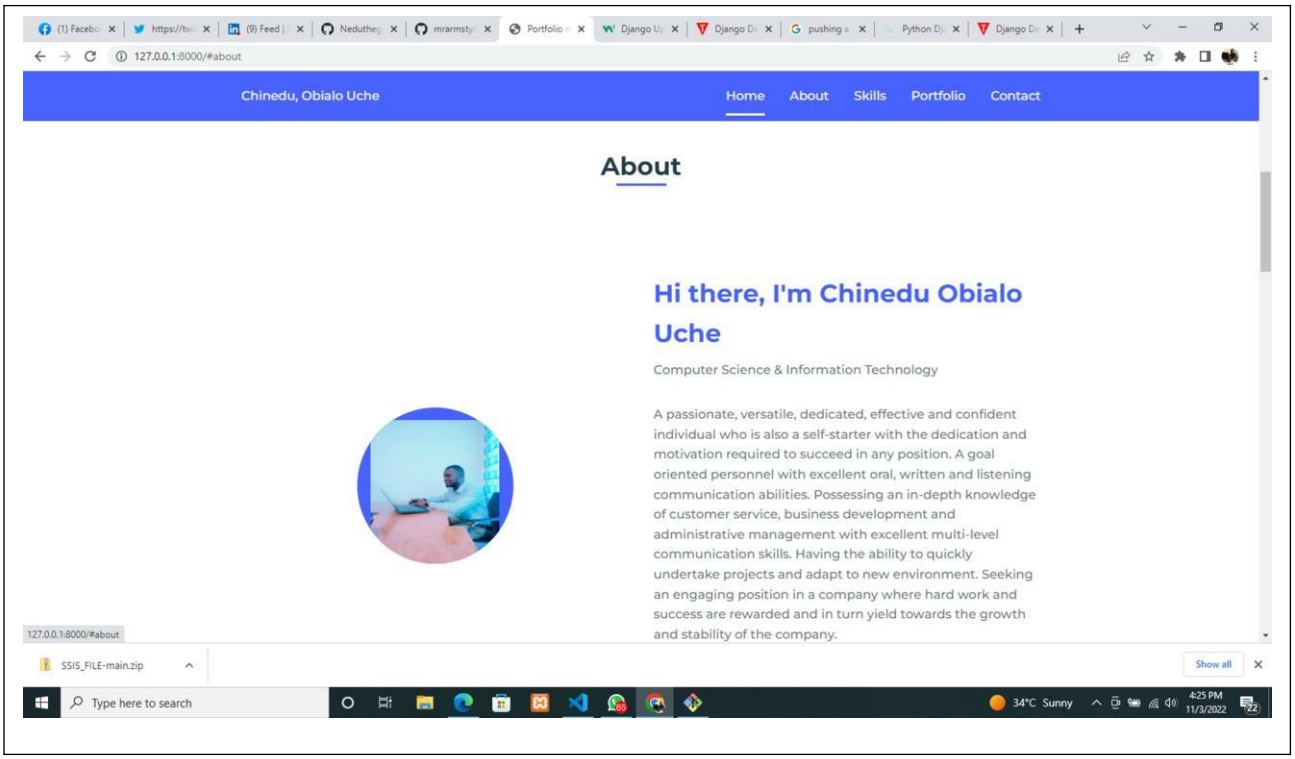
4.2 Website Design Screenshots

Figure 2 Homepage



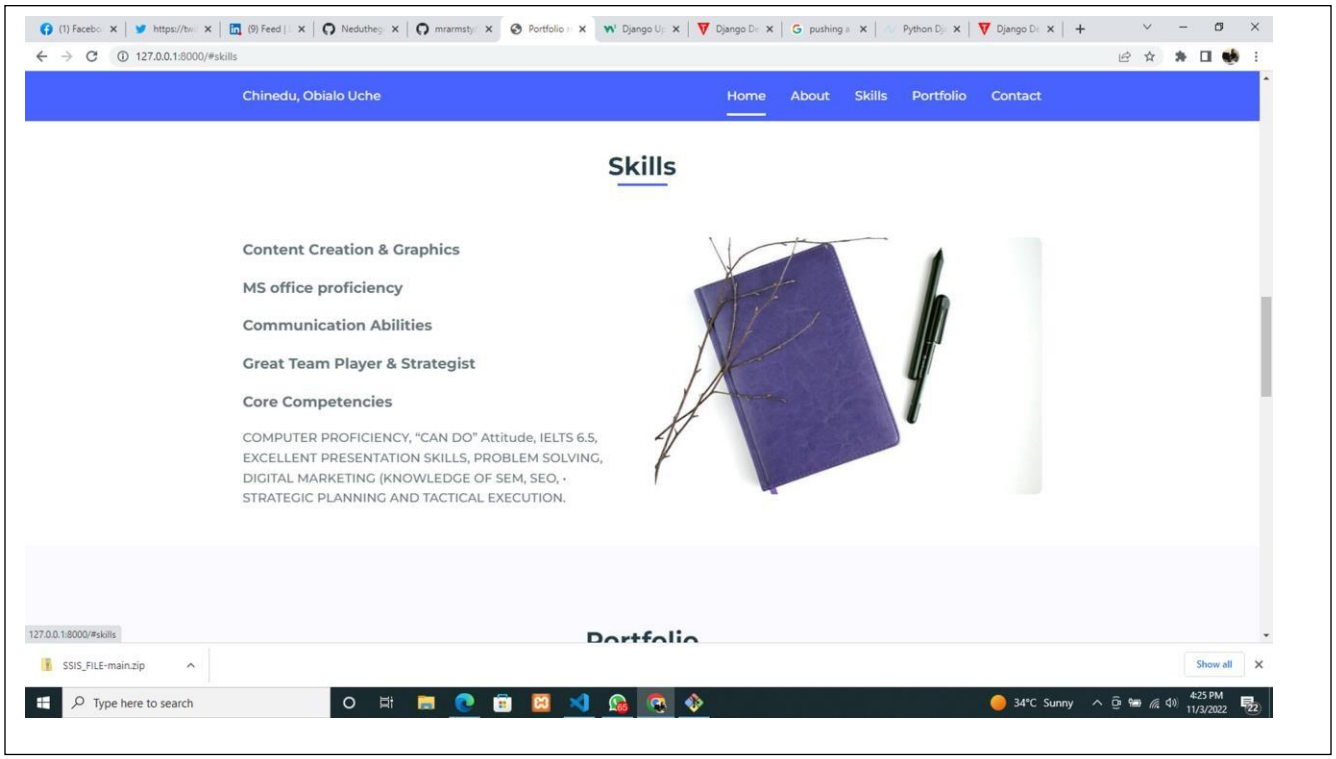
The homepage consists of four sections. Namely home, about, skills, portfolio and contact. The top bar of the homepage shows the user multiple details. It begins with “Hi there” because the page is merely welcoming the user into the site. To view the contents of each of the pages, it is required that the user clicks each of the item highlighted.

Figure 3 About page



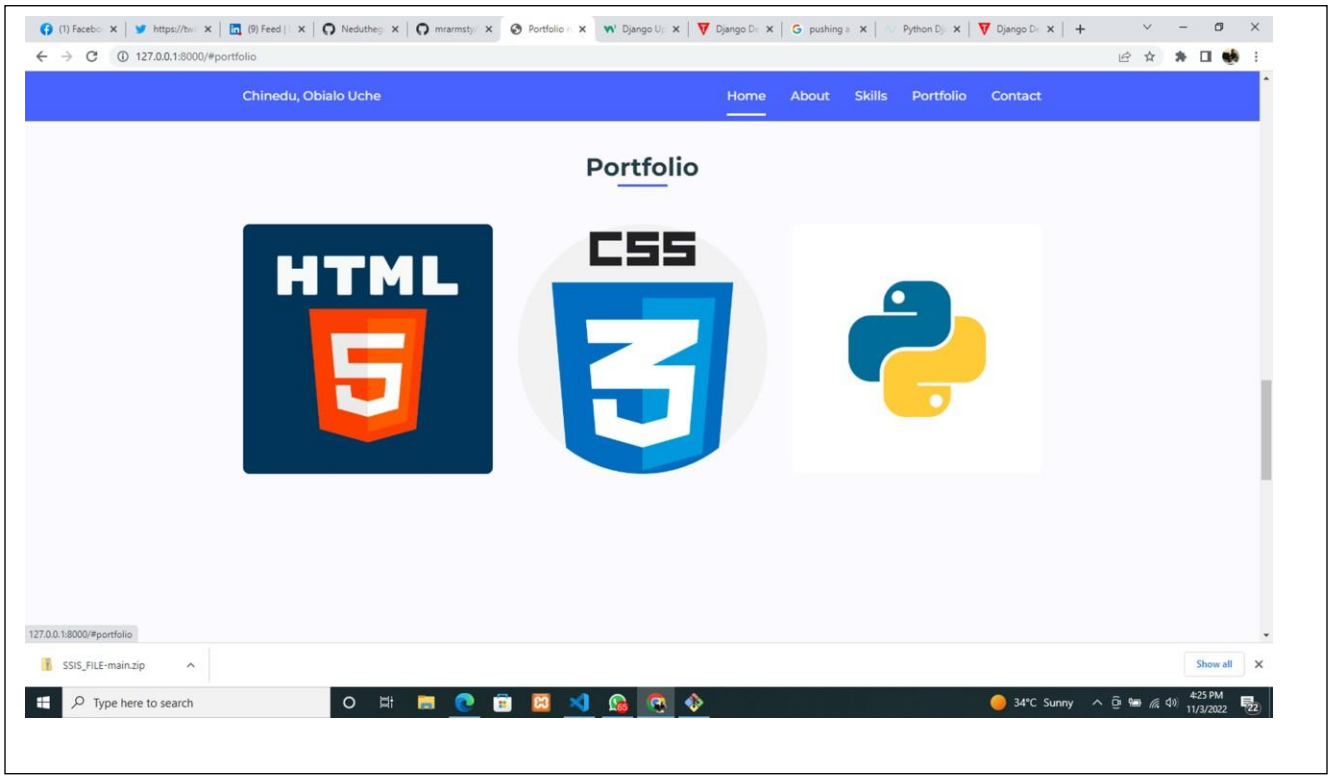
When about page is clicked, what appears is the information about the site owner, in this case Chinedu Obialo Uche. At this page, a brief information is presented about him. The reader will have to go through it to understand the personality of the site owner.

Figure 4 Skills page



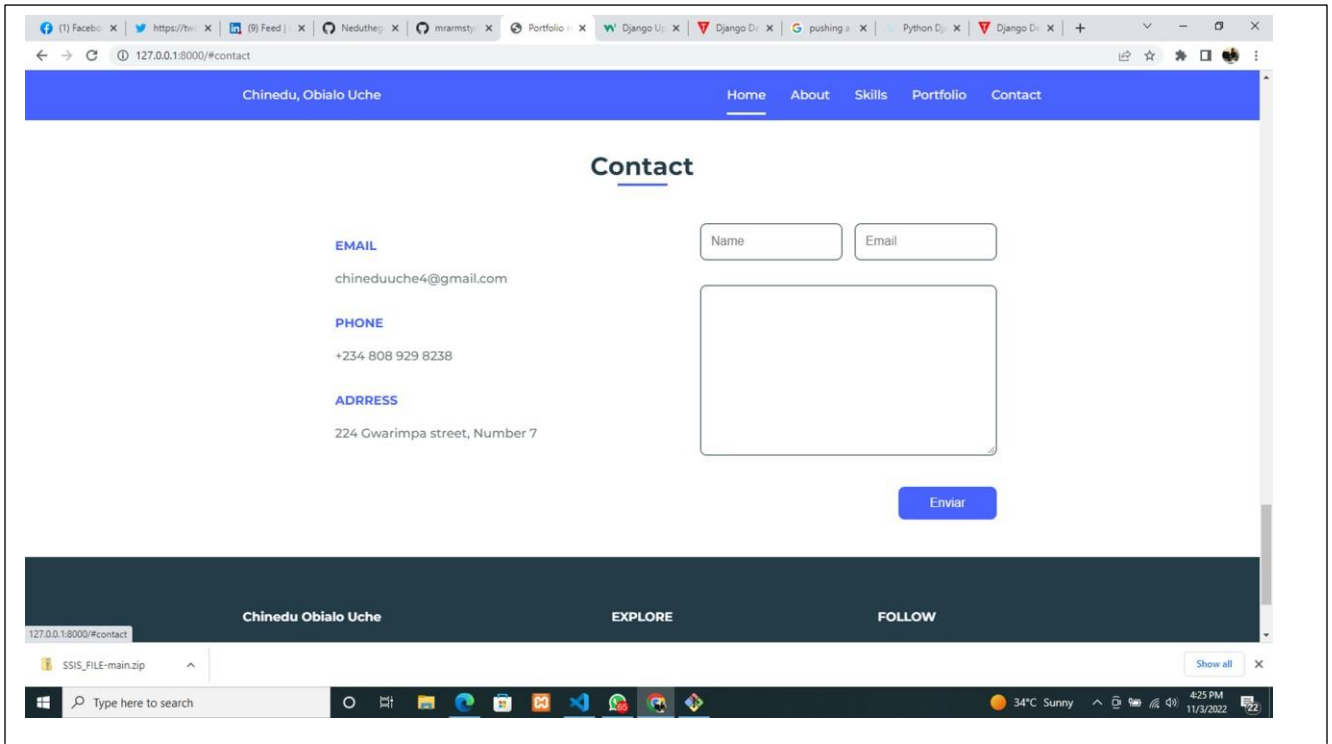
Under the skills page, the owner presents his expertise. As shown above, Chinedu Uche has skills in content creation and graphics, communication abilities, great team player and strategist. The core competencies of the owner is also presented.

Figure 5 Portfolio page



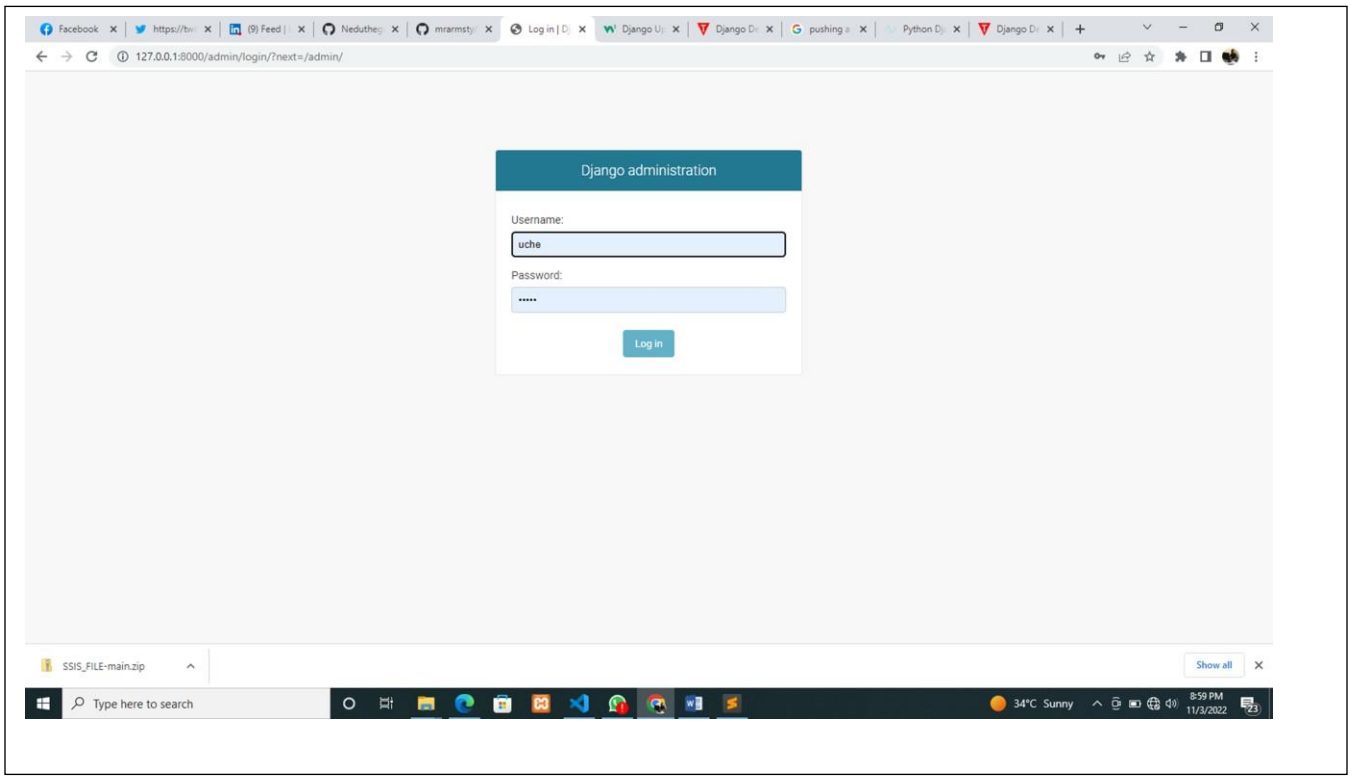
The portfolio page shows what is in store. As shown above, HTML and CSS are in store for use.

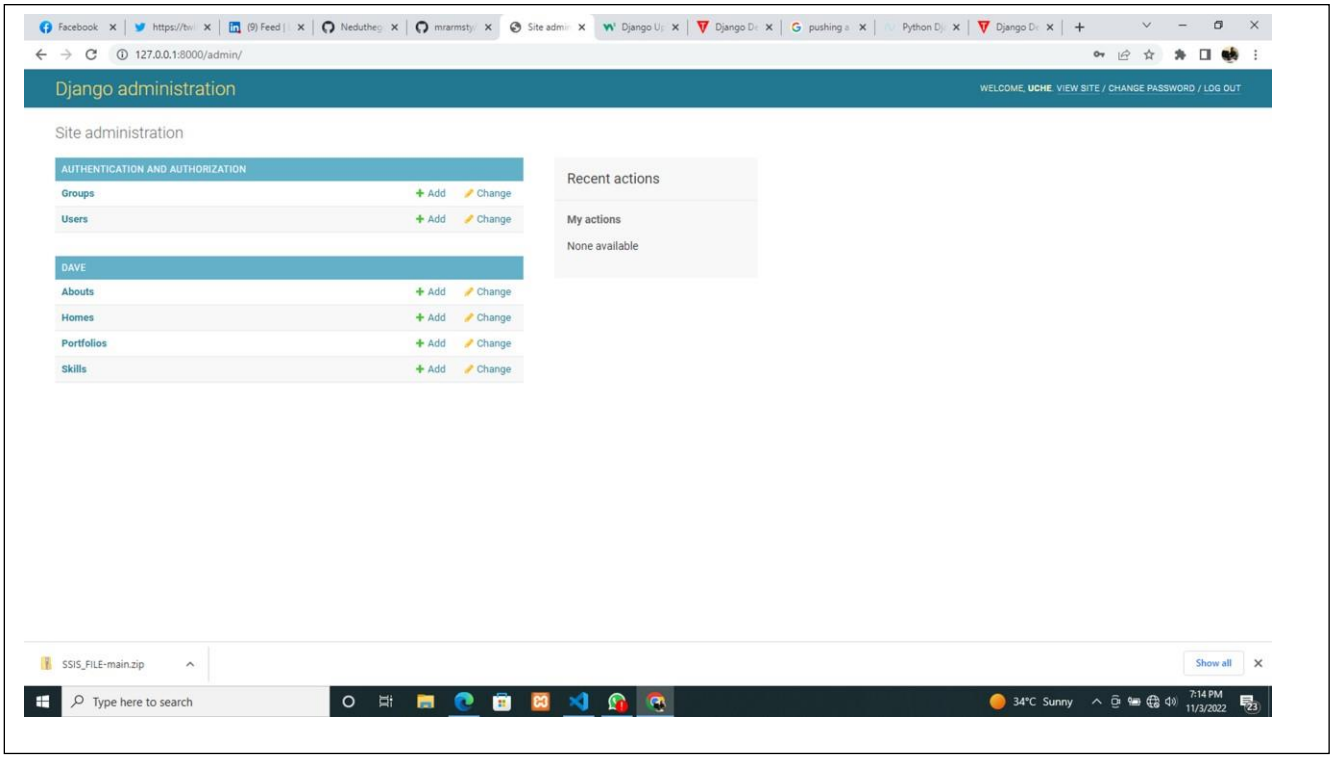
Figure 5: Contact page



The contact page is the page for information. Should anyone want to contact the Chinedu for a job based on his skills, the person will contact him either through the email, phone call or office and residential address. The person will have to type his name and his email address and state what the the enquiry is all about. The information can be sent to the owner by clicking the enter button.

Figure 6 Admin log in





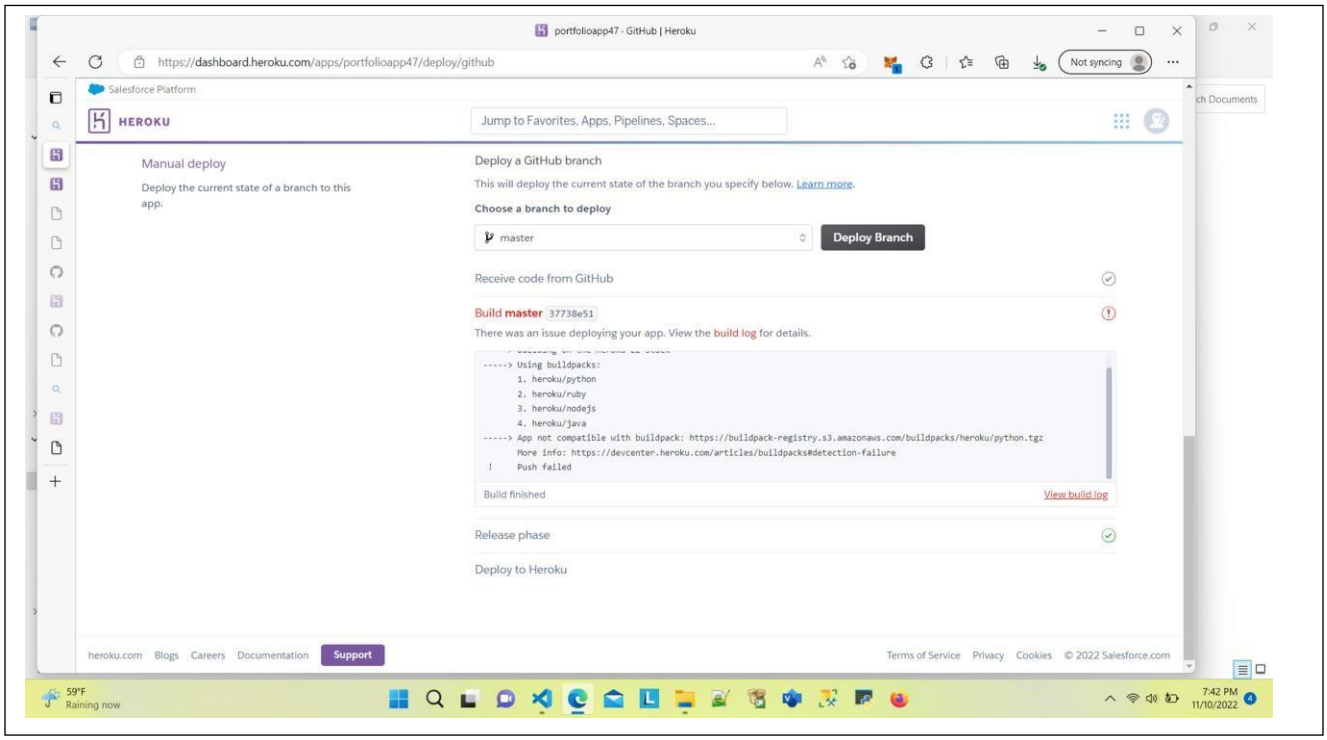
On this page, the admin logs in and can view all the data and add the data needed for the web application.

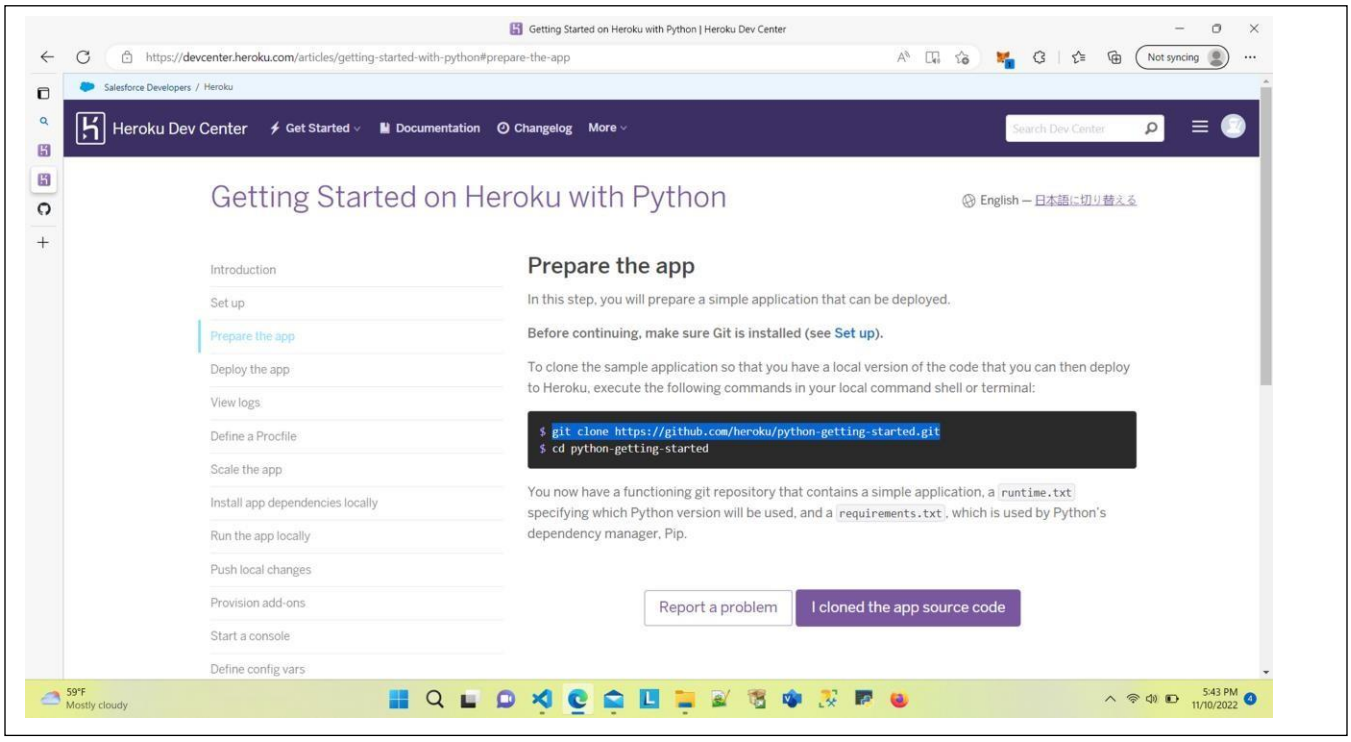
5 Heroku Deployment

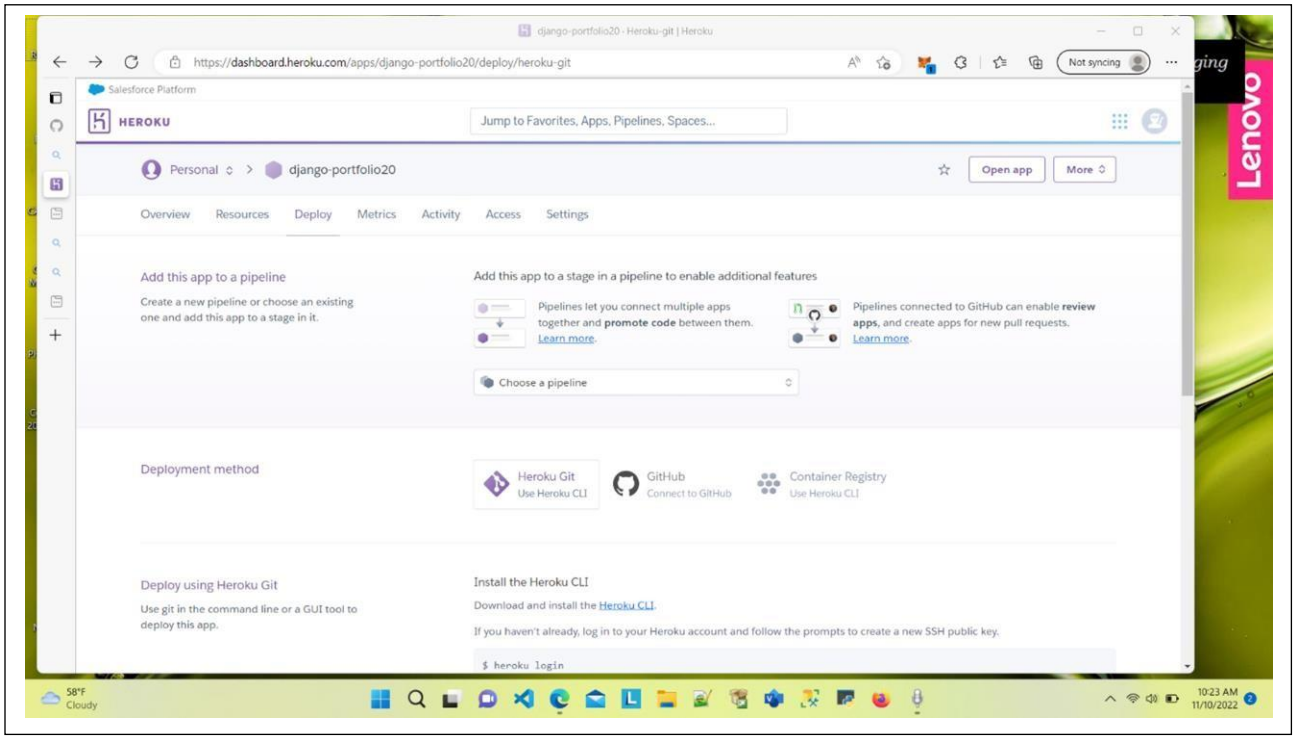
Heroku's utilities allowed me to deploy, run, and manage my website since it was in Python. The application's source code and dependency files provided enough information to build a website powered by the Heroku platform. The Heroku platform used Git as the primary means of deploying applications. Once the Heroku platform receives the application source, it will start building the source application. The build mechanism is usually language-specific, but follows the same pattern, usually taking the specified dependencies and building the required resources.

An application consists of source code, dependency descriptions, and a procfile. Procfiles lists process types for named commands that may be executed. Deploying your application involves submitting your application to Heroku via Git, GitHub, or an API. Buildpacks are behind the slug compilation process. A buildpack takes an application, its dependencies, and a language runtime and creates a slug. A slug is a ready-to-run bundle of sources, fetched dependencies, language runtime, and compiled/generated output of a build system. Configuration variables contain customizable configuration data that can be changed independently of your source code. Configuration is exposed to the running application via environment variables. Add-ons are specialized third-party cloud value-added services that can be easily attached to your application to extend its functionality. A release is a combination of slugs (applications), configuration variables, and addons.

Figure 7 Heroku deployment




Getting Started on Heroku with Python | Heroku Dev Center
https://devcenter.heroku.com/articles/getting-started-with-python#prepare-the-app
Not syncing
Salesforce Developers / Heroku
Heroku Dev Center
Get Started
Documentation
Changelog
More
Search Dev Center
Getting Started on Heroku with Python
English — 日本語に切り替える
Introduction
Set up
Prepare the app
Deploy the app
View logs
Define a Procfile
Scale the app
Install app dependencies locally
Run the app locally
Push local changes
Provision add-ons
Start a console
Define config vars
Prepare the app
In this step, you will prepare a simple application that can be deployed.
Before continuing, make sure Git is installed (see Set up).
To clone the sample application so that you have a local version of the code that you can then deploy to Heroku, execute the following commands in your local command shell or terminal:
$ git clone https://github.com/heroku/python-getting-started.git
$ cd python-getting-started
You now have a functioning git repository that contains a simple application, a runtime.txt specifying which Python version will be used, and a requirements.txt, which is used by Python's dependency manager, Pip.
Report a problem
I cloned the app source code
59°F
Mostly cloudy
5:43 PM
11/10/2022


django-portfolio20 · Heroku-git | Heroku
https://dashboard.heroku.com/apps/django-portfolio20/deploy/heroku-git
Not syncing
Salesforce Platform
HEROKU
Jump to Favorites, Apps, Pipelines, Spaces...
Personal
django-portfolio20
Open app
More
Overview
Resources
Deploy
Metrics
Activity
Access
Settings
Add this app to a pipeline
Create a new pipeline or choose an existing one and add this app to a stage in it.
Add this app to a stage in a pipeline to enable additional features
Pipelines let you connect multiple apps together and promote code between them.
Learn more.
Pipelines connected to GitHub can enable review apps, and create apps for new pull requests.
Learn more.
Choose a pipeline
Deployment method
Heroku Git
Use Heroku CLI
GitHub
Connect to GitHub
Container Registry
Use Heroku CLI
Deploy using Heroku Git
Use git in the command line or a GUI tool to deploy this app.
Install the Heroku CLI
Download and install the Heroku CLI.
If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.
$ heroku login
58°F
Cloudy
10:23 AM
11/10/2022
Lenovo

6. Security Threats and Measures

Cross-Site Request Forgery (commonly known as CSRF) allows attackers to trick users into doing things they do not want Web vulnerabilities. This allows an attacker to partially bypass the same-origin policy designed to prevent websites from interacting with each other.

6.1 How to Avoid CSRF Attack

To avoid CSRF attacks, A CSRF token is used. This is a unique secret, unpredictable value generated by a server-side application and sent to the client to be included in the client's next HTTP request. When a request is made later, the server-side program checks for the expected token and rejects it if the token is missing or invalid. By preventing an attacker from generating a perfectly valid HTTP request suitable for forwarding to the victim user, the CSRF token helps prevent CSRF attacks. Since the attacker cannot identify or predict the value of the user's CSRF token, he cannot generate a request containing all the parameters required for the application to accept the token.

7 Conclusion

Online channels like the Portfolio website make it simple for customers to contact site managers. Model-Template-Views (MTV) is an architectural design pattern used in the free and open source Django web framework for Python. The major objective of Django is to make it easier to build intricate database-driven websites. The flow diagram shows how the Django framework for python makes it possible for task to be easily performed by users of the site. The admin is also able to provide quick feedback to the users. The website was deployed to the heroku cloud deployment for ease of retrievability of data.

References

Baumgartner,P.and Malet, Y. 2015. *High performance Django* (1st ed.), Lincoln Loop, p. 184
Django (2022). General. Available at: https://docs.djangoproject.com/en/dev/faq/general/#django-appears-to-be-a-mvc-framework-but-you-call-the-controller-the-view-and-the-view-the-template-how-come-you-don-t-use-the-standard-names [Accessed on 9 November, 2022]

Holovaty, A. Kaplan-Moss, J. 2013. *The Django Book. Retrieved 3 September 2013*. Django follows this MVC pattern closely enough that it can be called an MVC framework. Available at https://pdfprodocs.vip/download/4330427-the-definitive-guide-to-django-holovaty-adrian-kaplan-moss-jacob [accessed on 9 November, 2022]

Links:

GitHub Link: https://github.com/Neduthegoat/Django-portfolio47

Note: Heroku Link: Unfortunately, I met some hurdles during the uploading of my work on Heroku. I tried to fix it up but unfortunately I couldn't and needed to beat the deadline.

Attached are some of the screenshots to the error message I got from Heroku. See Figure 7