Cloud Web Application with CI/CD

A comprehensive report on hosting a Health Diagnostic Application in the Cloud with Continuous Integration and Continuous Deployment

Student:

Srikanth Shilesh Pasam (10387794)

Teacher:

Obinna Izima

Course:

Web Development for Information Systems (B9IS109_1920_TMD2) CA 2

Index

List of Illustrations

Introduction

This CA aims to research and develop a web application hosted in the cloud. The concept of web application is based on the idea of a health diagnostic tool. This tool will help users track and monitor their heart health.

The current scope of the project is only limited to users manually providing their medical data post, in which the analysis is done. It is a medical fact that a stroke can be detected even before it occurs. Based on that, this scope can be extended in the future by allowing automatic data collection and continuous monitoring through the use of microsensors placed on the user's body. The data can be relayed to the application on the user's phone or smartwatch. By continuously monitoring the user's heart, medical complications such as strokes can be avoided, thereby saving lives.

Research & Planning

To develop the application, the project is divided into three parts. The front end involves the UI/UX, the backend, which includes the database, cloud, and CI/CD and a framework to enable communication between the two. Based on that, the following list of technologies will be used:

- HTML5
- CSS3
- jQuery
- Bootstrap
- Python Django Framework
- SQL Lite 3
- Heroku Cloud Application Platform
- Git Hub

HTML5

HTML stands for HyperText Markup Language. It is used to describe the structure of a webpage. HTML elements tell the browser how to display a webpage. HTML is written using tags. These tags are used by the browsers when displaying a webpage.
HTML5 is the latest evolution of the standard that defines HTML. The term represents two different concepts. It is a new version of the language HTML, with new elements, attributes, and behaviors, and a broader set of technologies that allows the building of more diverse and influential Web sites and applications (*HTML5*, 2019, p. 5).

CSS3

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.
CSS is one of the core languages of the open Web and is standardized across Web browsers according to the W3C specification (*CSS: Cascading Style Sheets*, 2019).

jQuery

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript (js.foundation, no date).

Bootstrap

Bootstrap is an open-source toolkit for developing with HTML, CSS, and JS. Using Bootstrap, one can quickly prototype their ideas or build an entire app with their Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery (Mark Otto, no date).

Django Framework

A framework is a software that organizes the architecture of an application making development more straightforward. Django is a web framework that uses Python to create websites. The structure uses the MVC pattern. MVC – Models-Views-Controllers represents the three application layers. This pattern helps separate logic from representation.

Models - These represent data organization in a database. In simple words, we can say that each model defines a table in the database and the relations between other models. It's thanks to them that every bit of data is stored in the database.

Views - These contain all the information that will be sent to the client. They make views that the final HTML document will generate. We can associate the HTML code with the views.

Controllers - These contain all the actions performed by the server and are not visible to the client. The controller checks whether the user is authenticated, or it can generate the HTML code from a template.
(Dauzon, Bendoraitis, and Ravindran, 2016, p. 6)

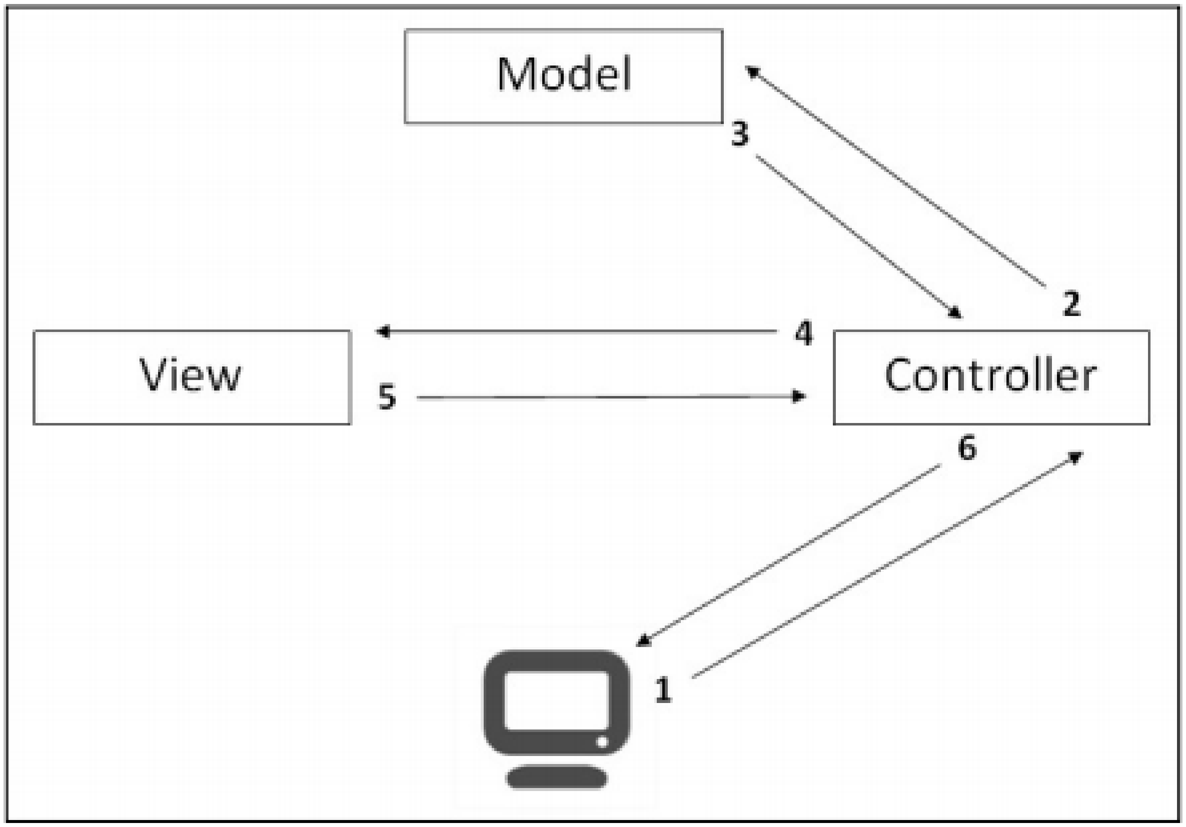


Figure 1-Django MVC (Dauzon, Bendoraitis, and Ravindran, 2016, p. 6)

The following are the steps that are developed in an application with the MVC pattern:

1. The client sends a request to the server asking to display a page.
2. The controller uses a database through models. It can create, read, update, or delete any record or apply any logic to the retrieved data.
3. The model sends data from the database.
4. The controller injects data into a view to generate it.
5. The view returns its content, depending on the data given by the controller.
6. The controller returns the HTML content to the client.

   (Dauzon, Bendoraitis, and Ravindran, 2016, p. 7)

Django follows the Don't Repeat Yourself (DRY) principle, making this framework time-efficient. In other words, there's no need to rewrite existing code because Django allows you to assemble your website like a Lego set. The framework is well-suited for high load systems and can decrease development time thanks to lots of helper objects (Zublenko, no date).

SQL Lite 3

SQLite is a public-domain software package that provides a relational database management system or RDBMS. Relational database systems are used to store user-defined records in large tables. In addition to data storage and management, a database engine can process complex query commands that combine data from multiple tables to generate reports and data summaries.

SQLite is defined by the following features:

- Serverless - SQLite does not require a separate server process or system to operate. The SQLite library accesses its storage files directly.
- Zero Configuration - No server means no setup. Creating an SQLite database instance is as easy as opening a file.
- Cross-Platform - The entire database instance resides in a single cross-platform file, requiring no administration.
- Self-Contained - A single library contains the entire database system, which integrates directly into a host application.
- Small Runtime Footprint - The default build is less than a megabyte of code and requires only a few megabytes of memory.
- Transactional - SQLite transactions are fully ACID-compliant, allowing safe access from multiple processes or threads.
- Full-Featured - SQLite supports most of the query language features found in the SQL92 (SQL2) standard.
- Highly Reliable - The SQLite development team takes code testing and verification very seriously.

> Overall, SQLite provides a very functional and flexible relational database environment that consumes minimal resources and creates a minimal hassle for developers and users (Kreibich, 2010, p. 2).

Heroku Cloud Platform

Heroku is a platform as a service based on a managed container system, with integrated data services and a robust ecosystem, for deploying and running modern apps Heroku runs your apps inside dynos — smart containers on a reliable, fully managed runtime environment (*Platform as a Service | Heroku*, no date).

GitHub

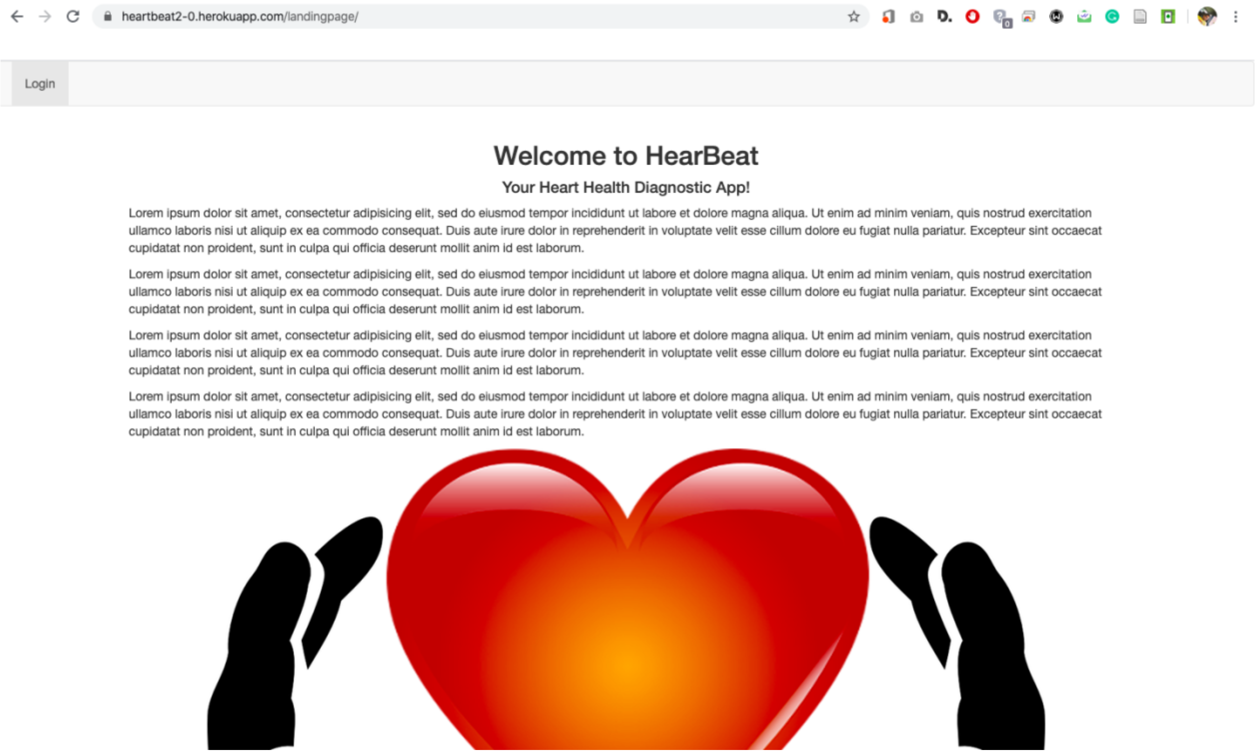
GitHub is a Git repository hosting service primarily used for version control. While Git is a command-line tool, GitHub provides a Web-based graphical interface. It also provides access control and several collaboration features, such as necessary task management tools for every project (‘What Exactly Is GitHub Anyway?’, 2012).

Project Overview

UX Design

The core value of UX for the project is simplicity. The primary motivation behind this being the target audience. As heart conditions are more common among older age groups, the UX of the web app is designed to be simple and straight forward, thus eliminating any unnecessary confusion.

There are four pages in the application. The home page welcomes the user and gives a brief introduction to the application. From here, the user can choose to log in or signup. After creating an account, the user will be logged in to the application. Here they are given the fields which need to be filled. Once all the fields have been filled, the form can be submitted. On submission, the application will reference the user input with its medical database to validate the heart condition of the user. The results are displayed on the final page in the form of graphs.



heartbeat2-0.herokuapp.com/landingpage/

Welcome to HearBeat

Your Heart Health Diagnostic App!

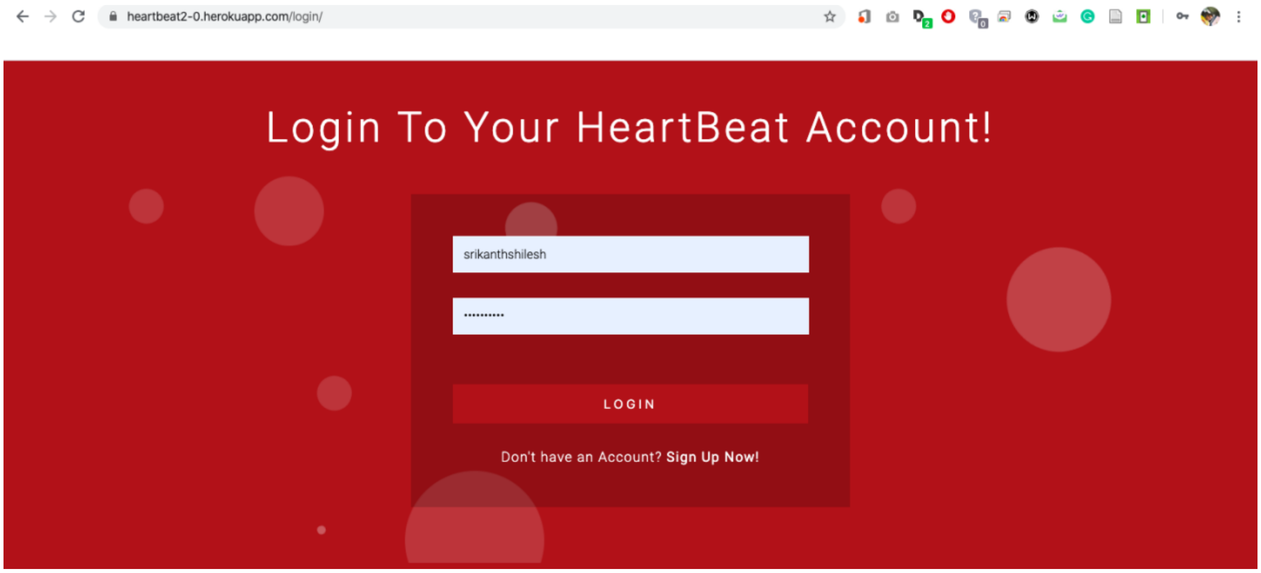
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Figure 2 - Landing Page



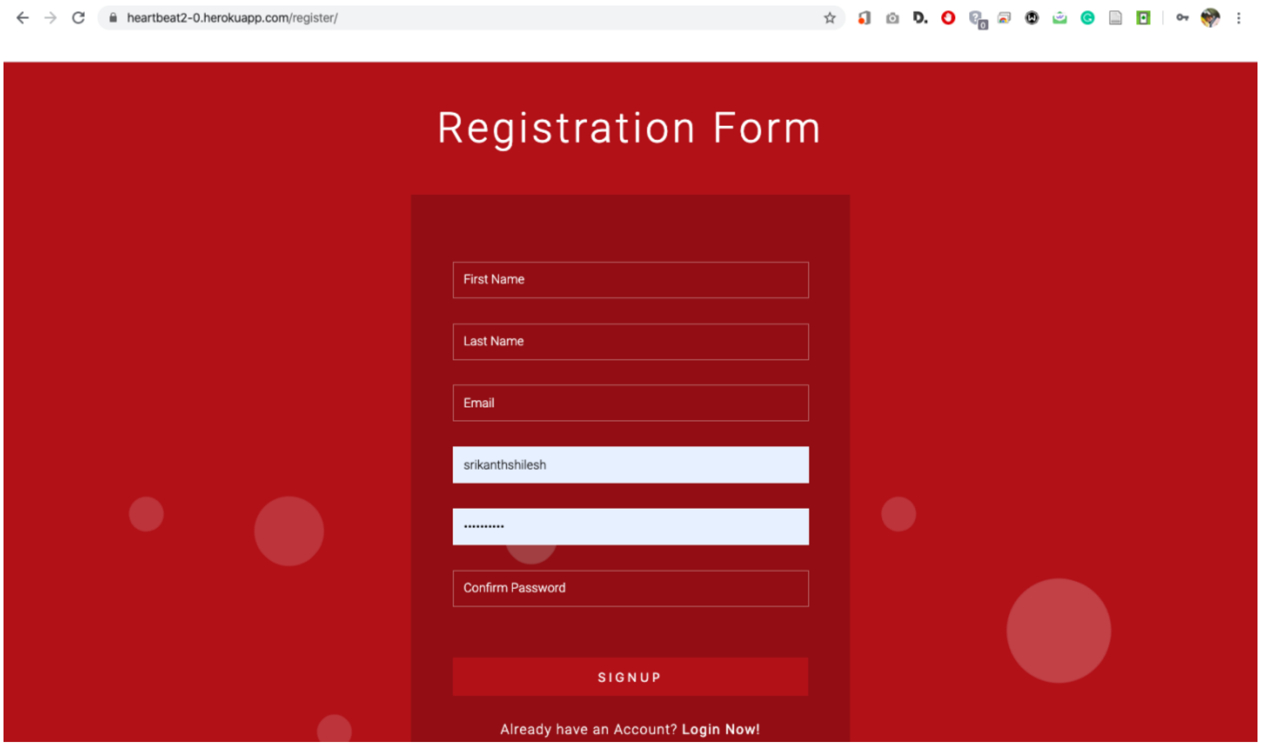
heartbeat2-0.herokuapp.com/login/

Login To Your HeartBeat Account!

srikanthshilesh

••••••••••

Figure 3 - Login Page



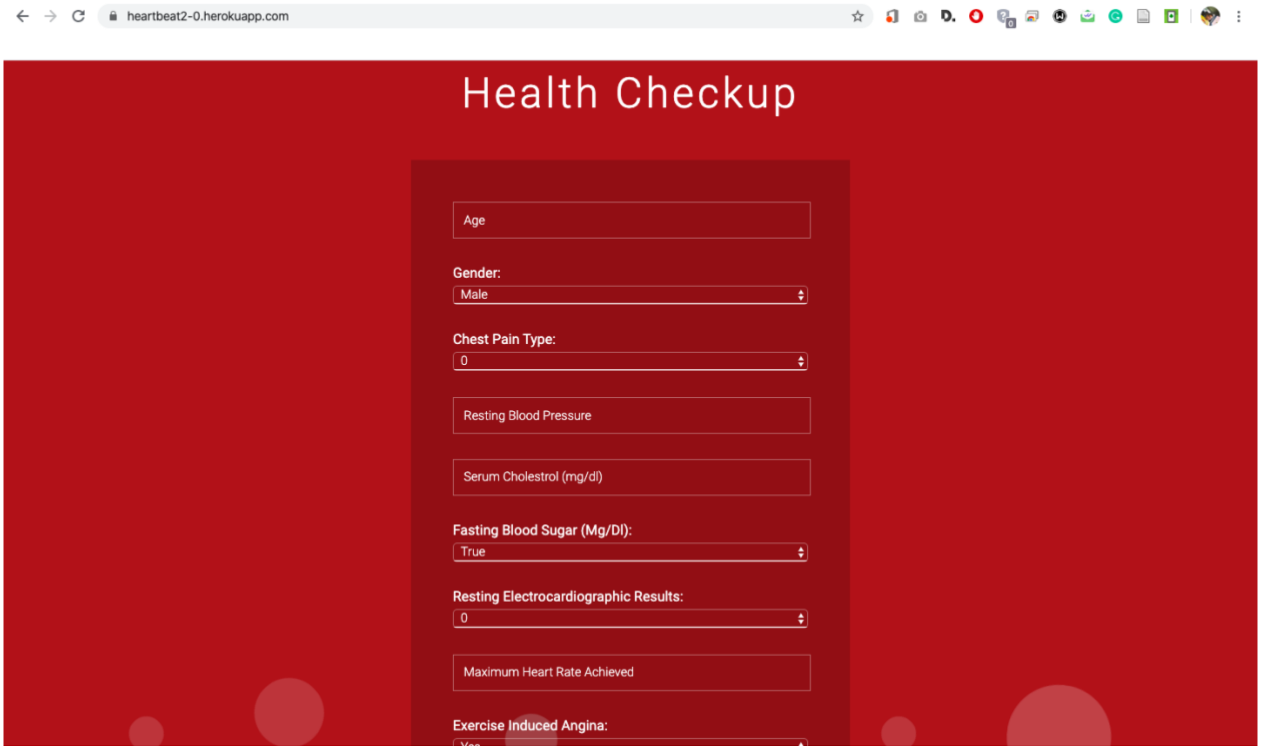
heartbeat2-0.herokuapp.com/register/

Registration Form

srikanthshilesh

••••••••••

Figure 4 - Registration Page



heartbeat2-0.herokuapp.com

Health Checkup

Gender: Male

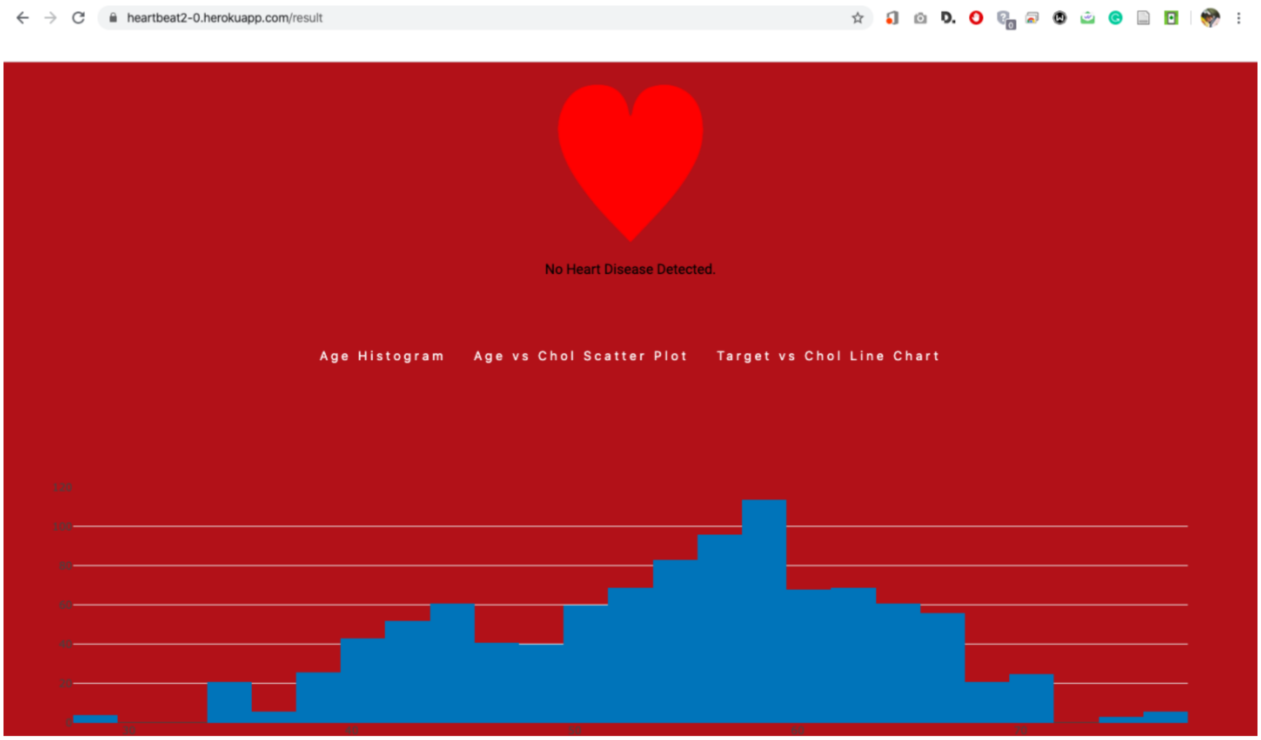
Chest Pain Type: 0

Fasting Blood Sugar (Mg/Dl): True

Resting Electrocardiographic Results: 0

Exercise Induced Angina: Yes

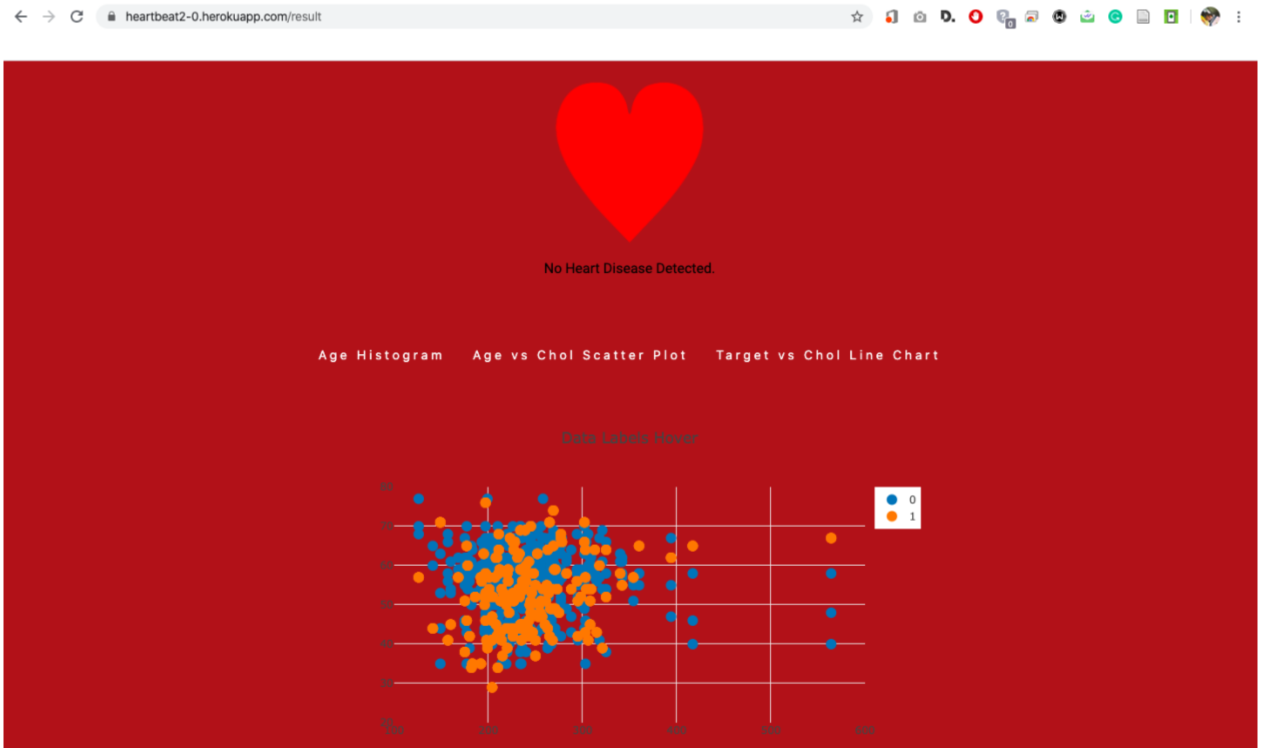
Figure 5 - Form Entry Page



No Heart Disease Detected.

Age Histogram Age vs Chol Scatter Plot Target vs Chol Line Chart

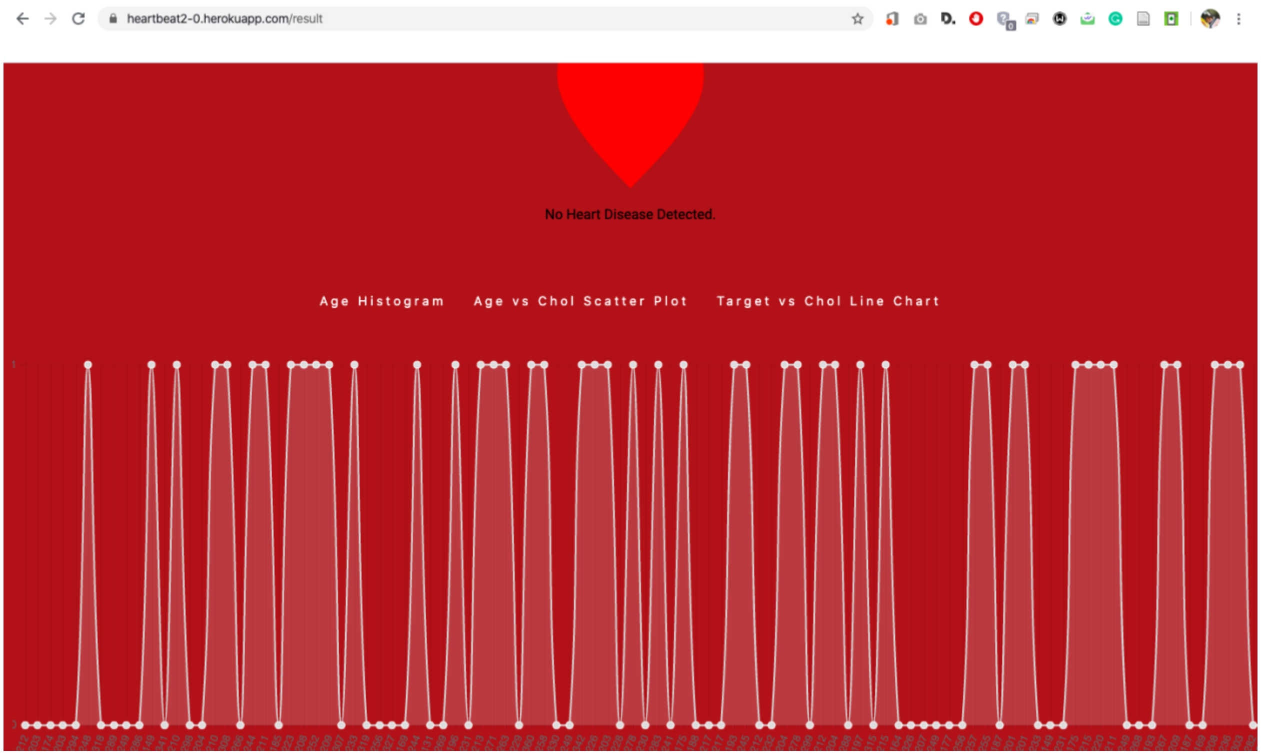
Figure 6 - Age Result Graph



No Heart Disease Detected.

Age Histogram Age vs Chol Scatter Plot Target vs Chol Line Chart

Figure 7 - Age vs. Cholesterol Result Graph



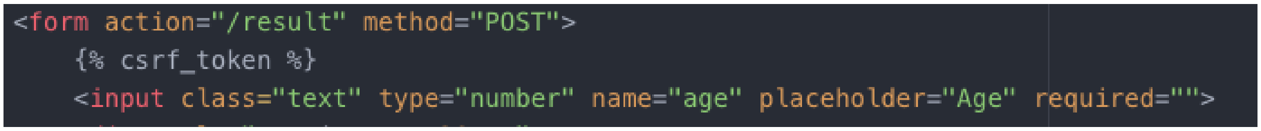
No Heart Disease Detected.

Figure 8 - Cholesterol Result Line Graph

Security

The following security features have been implemented in the application:

1. Cross-Site Request Forgery (CSRF) Protection – CSRF works by checking for a secret in each POST request. This ensures that a malicious user cannot "replay" a form POST to your website and have another logged in user unwittingly submit that form. The malicious user would have to know the secret, which is user-specific (using a cookie).



```
<form action="/result" method="POST">
    {% csrf_token %}
    <input class="text" type="number" name="age" placeholder="Age" required="">
```

Figure 9 - CSRF Token

2. Clickjacking Protection - Django contains clickjacking protection in the form of the X-Frame-Options middleware, which is a supporting browser that can prevent a site from being rendered inside a frame.

3. Host Header Validation - Django uses the Host header provided by the client to construct URLs. This can be set in the settings.py file in the allowed host's line.

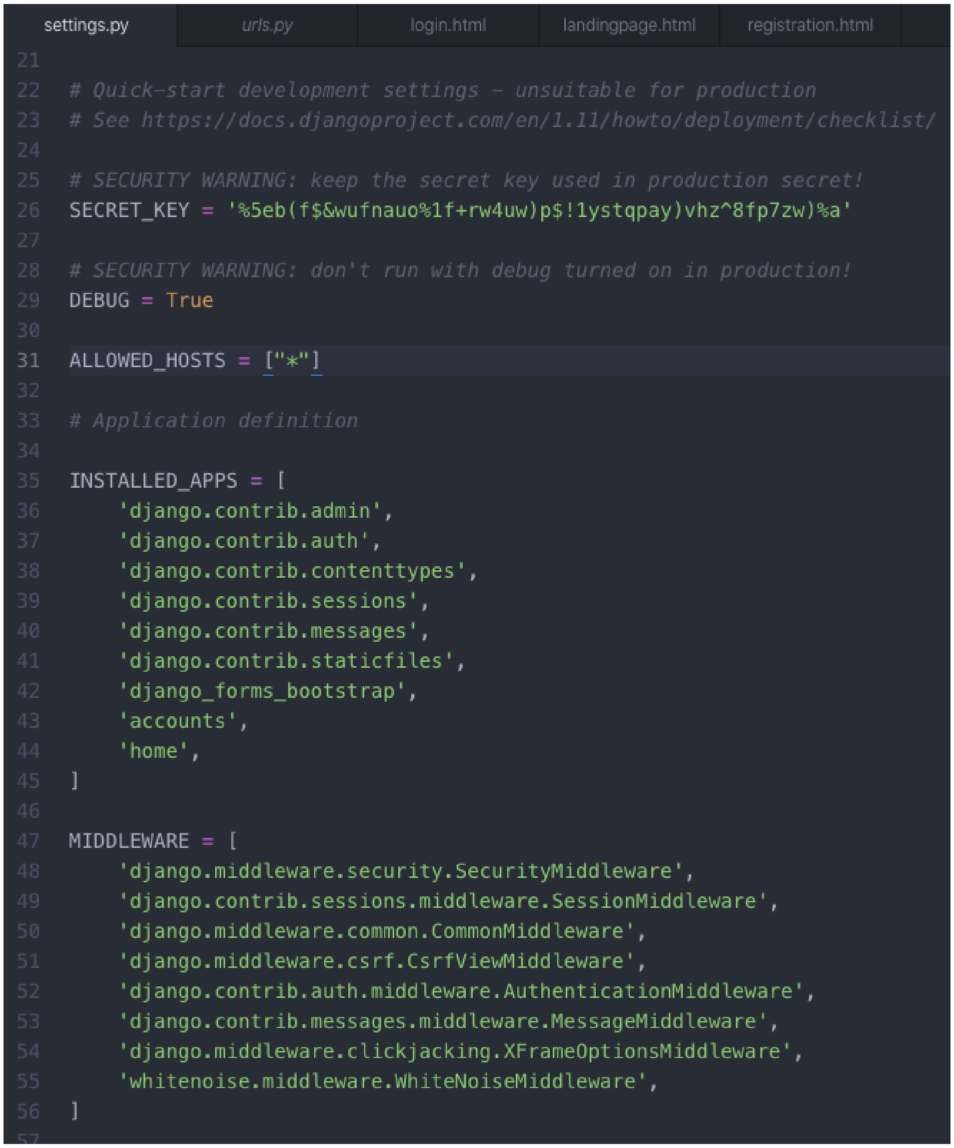
4. SQL Injection Protection - Django's querysets are protected from SQL injection since their queries are constructed using query parameterization. A query's SQL code is defined separately from the query's parameters. Since parameters may be user-provided and, therefore, unsafe, they are escaped by the underlying database driver.

5. Cross-Site Scripting (XSS) Protection - Django templates protect against the majority of XSS attacks though it has certain limitations concerning HTML and Database interactions.

6. Session Security – Every session created for the user is tracked and managed in the database for security. This is done using django.contrib. Sessions command.

7. SSL/HTTPS – This feature is provided by Heroku Cloud services hosting the app in an HTTPS server.

   (*Security in Django | Django documentation | Django*, no date)

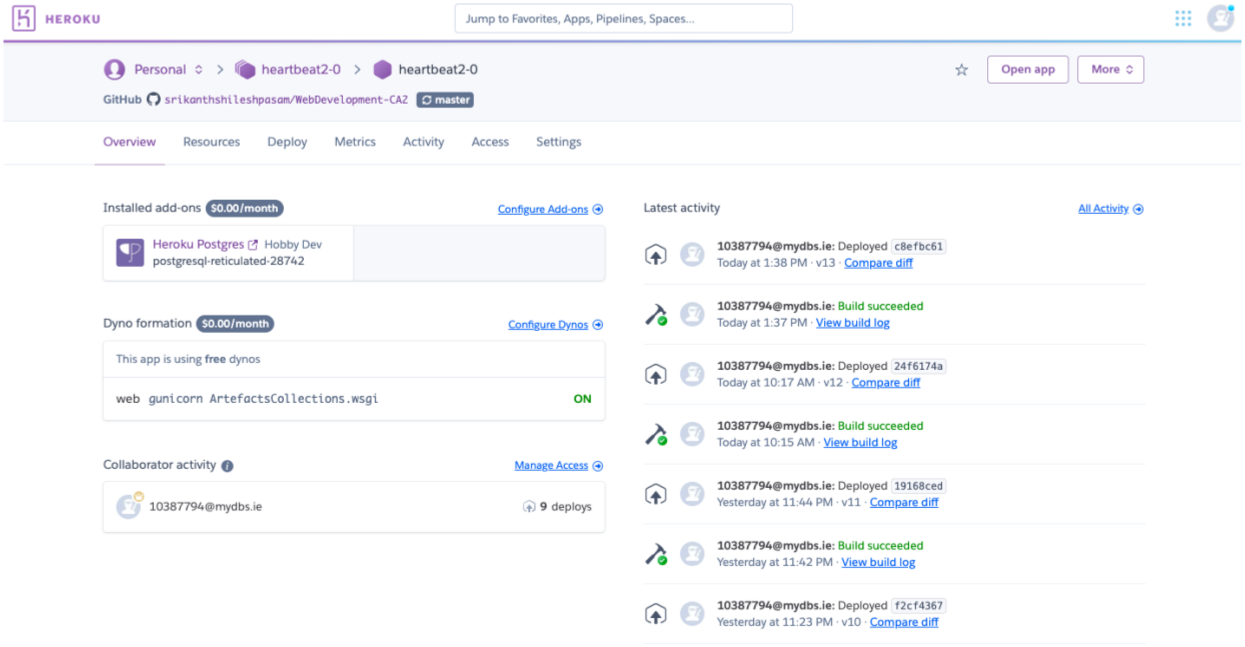


```

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/1.11/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '%5eb(f$&wufnauo%1f+rw4uw)p$!1ystqpay)vhz^8fp7zw)%a'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ["*"]

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'django_forms_bootstrap',
    'accounts',
    'home',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
    'whitenoise.middleware.WhiteNoiseMiddleware',
]
```

Figure 10 - Security Parameters

Project Deployment

The project is deployed using the CI/CD model. The continuous integration and continuous deployment features are enabled using GitHub for managing the source code. Whenever a new commit is pushed, Heroku builds a new version with the updated files.



GitHub srikanthshileshpasam/WebDevelopment-CA2 master

Installed add-ons $0.00/month

Heroku Postgres Hobby Dev
postgresql-reticulated-28742

Dyno formation $0.00/month

This app is using **free** dynos

`web gunicorn ArtefactsCollections.wsgi` ON

Collaborator activity

10387794@mydbs.ie 9 deploys

Latest activity

10387794@mydbs.ie: Deployed `c8efbc61`
Today at 1:38 PM · v13 · Compare diff

10387794@mydbs.ie: Build succeeded
Today at 1:37 PM · View build log

10387794@mydbs.ie: Deployed `24f6174a`
Today at 10:17 AM · v12 · Compare diff

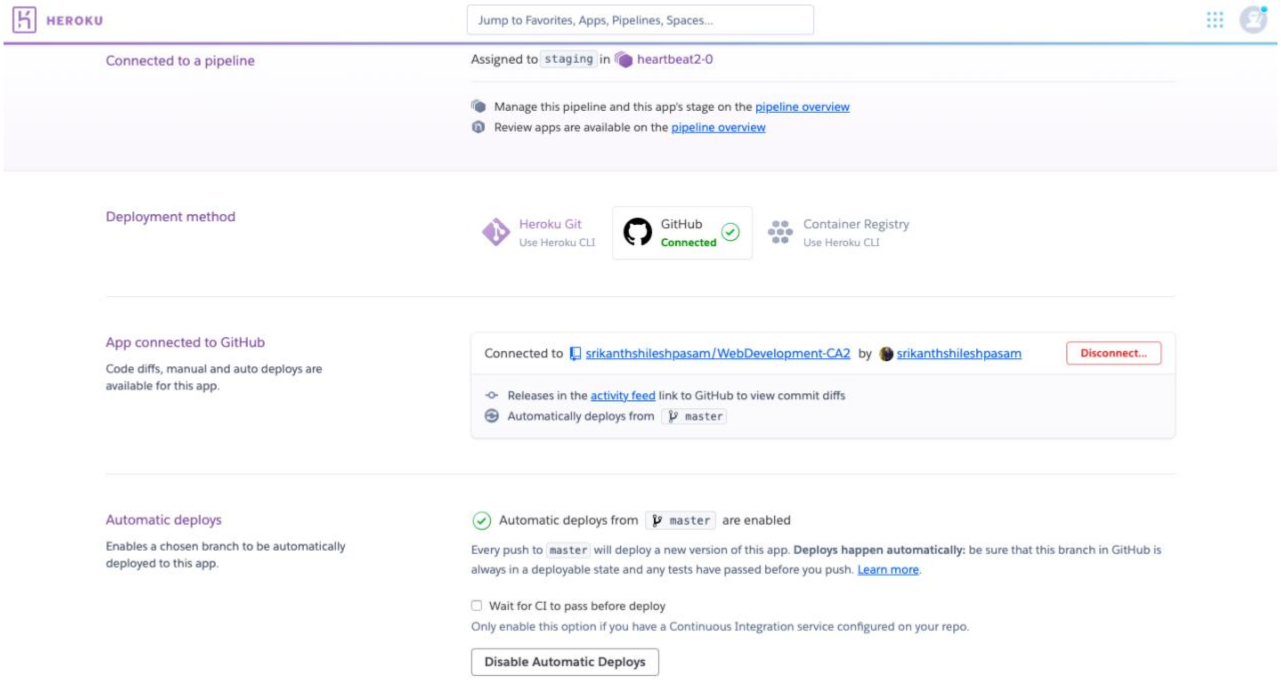
10387794@mydbs.ie: Build succeeded
Today at 10:15 AM · View build log

10387794@mydbs.ie: Deployed `19168ced`
Yesterday at 11:44 PM · v11 · Compare diff

10387794@mydbs.ie: Build succeeded
Yesterday at 11:42 PM · View build log

10387794@mydbs.ie: Deployed `f2cf4367`
Yesterday at 11:23 PM · v10 · Compare diff

Figure 11 - Heroku Deployment



Connected to a pipeline

Assigned to `staging` in heartbeat2-0

Manage this pipeline and this app's stage on the pipeline overview

Review apps are available on the pipeline overview

Deployment method

App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to srikanthshileshpasam/WebDevelopment-CA2 by srikanthshileshpasam

Releases in the activity feed link to GitHub to view commit diffs

Automatically deploys from `master`

Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

✓ Automatic deploys from `master` are enabled

Every push to `master` will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch in GitHub is always in a deployable state and any tests have passed before you push. Learn more.

□ Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

Figure 12 - Heroku and GitHub CI/CD

Conclusion

The project has been successfully developed and hosted in the cloud. Users can access the application using the link given in the Appendix. The user can register for a new account and then be able to log in. Post logging in, the user can enter their medical data and submit the form. This medical data is referenced against the medical records in the application database, and the result is displayed to the user.

Reflection

The reason I chose this idea for my project is connected to my Teach For India Fellowship and my students whom I taught back then. Very recently, one of my student's father passed away due to a heart attack. He was the only bread earner for his family which is below the poverty line, and this incident has had a drastic impact on their family. The family has five children, and now all of them have dropped out of school. I am trying to raise funds to help the family, but this is only a temporary solution. I spoke to the kid, and she told me that her father was complaining about pain in his chest for many months but only thought it was due to his intensive physical labor and hence ignored it. Thinking about this has made me realize that if he could have had been warned earlier, this could have been avoided. This has been my motivation to build the application.

I come from an electronics background with no prior experience in the field of computer science. Initially, I was very skeptical about my choice to pursue a Masters in this field. Six months down the line to present, I am so much more confident in my abilities. Working on this project has been the best learning experience. The practical applications of this has given me a lot of motivation to build this into a career path post my Masters. I still have much more to learn, but this CA has helped build excellent foundation skills.

Bibliography

CSS: Cascading Style Sheets (2019) *MDN Web Docs*. Available at: https://developer.mozilla.org/en-US/docs/Web/CSS (Accessed: 13 March 2020).

Dauzon, S., Bendoraitis, A. and Ravindran, A. (2016) *Django: Web Development with Python*. Packt Publishing Ltd.

HTML5 (2019) *MDN Web Docs*. Available at: https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5 (Accessed: 13 March 2020).

js.foundation, J. F.- (no date) ‘jQuery’. Available at: https://jquery.com/ (Accessed: 13 March 2020).

Kreibich, J. A. (2010) *Using SQLite: Small. Fast. Reliable. Choose Any Three.* O’Reilly Media, Inc.

Mark Otto, J. T., and Bootstrap (no date) *Bootstrap*. Available at: https://getbootstrap.com/ (Accessed: 13 March 2020).

Platform as a Service | Heroku (no date). Available at: https://www.heroku.com/platform (Accessed: 13 March 2020).

Security in Django | Django documentation | Django (no date). Available at: https://docs.djangoproject.com/en/3.0/topics/security/ (Accessed: 13 March 2020).

‘What Exactly Is GitHub Anyway?’ (2012) *TechCrunch*, 14 July. Available at: http://social.techcrunch.com/2012/07/14/what-exactly-is-github-anyway/ (Accessed: 13 March 2020).

Zublenko, E. (no date) ‘Why Django is the Best Web Framework for Your Project’. Available at: https://steelkiwi.com/blog/why-django-best-web-framework-your-project/ (Accessed: 13 March 2020).

Appendix

GitHub repository link:

https://github.com/srikanthshileshpasam/WebDevelopment-CA2

Heroku link:

https://heartbeat2-0.herokuapp.com/landingpage/