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

[Index 3](#_Toc34987854)

[List of Illustrations 4](#_Toc34987855)

[Introduction 5](#_Toc34987856)

[Background 5](#_Toc34987857)

[Cloud Computing 5](#_Toc34987858)

[Automation 5](#_Toc34987859)

[DNS 5](#_Toc34987860)

[Project Overview 5](#_Toc34987861)

[Technical Description 5](#_Toc34987862)

[GitHub 5](#_Toc34987863)

[Domain Registration 5](#_Toc34987864)

[Route 53 5](#_Toc34987865)

[Testing 6](#_Toc34987866)

[Conclusion 6](#_Toc34987867)

[Proof of Concept 6](#_Toc34987868)

[Reflection 6](#_Toc34987869)

[Bibliography 7](#_Toc34987870)

[Appendix 1 7](#_Toc34987871)

[Appendix 2 7](#_Toc34987872)

# List of Illustrations

# Introduction

The aim of this CA is to research and develop a web application hosted in the cloud. The concept of the web application is based on the idea of a health diagnostic tool. This tool will help users track and monitor their heart health. Users will be able to login to their secure account in the application. They will then be able to provide the various medical readings taken from an ECG. The application will then cross reference this with its medical database and suggest the user if they need to act on it. By regularly monitoring their heart health, the users will be able to avoid any life-threatening complications before they occur.

The current scope of the project is only limited to users manually providing their medical data post 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 micro sensors placed on the user’s body. The data can be relayed to the application on the user’s phone or smart watch. By continuously monitoring the user’s heart, medical complications such as strokes can be avoided thereby saving lives.

# Research & Planning

In order to develop the application, the project is divided into three parts. The front end which 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 Hyper Text 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 larger set of technologies that allows the building of more diverse and powerful 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 easier. Django is a web framework that uses Python to create websites. The framework uses the MVC pattern. MVC – Models-Views-Controllers represents the three application layers. This patters 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)

A screenshot of a cell phone

Description automatically generated

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

The following are the steps that are followed 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; for example, it sends a product list if we have an online shop.
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. With some adjustments, both the library size and memory use can be significantly reduced.
* 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 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 powerful 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 basic 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 login or signup. After creating an account, the user will be logged in to the application. Here they are given the fields which needs 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 in order to validate the heart condition of the user. The results are displayed in the final page in the form of graphs.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

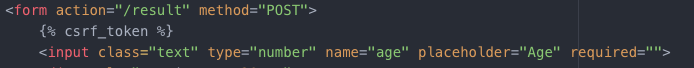
A screenshot of a cell phone

Description automatically generated

## 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).



1. Clickjacking Protection - Django contains [clickjacking protection](https://docs.djangoproject.com/en/3.0/ref/clickjacking/#clickjacking-prevention) in the form of the X-Frame-Options middleware which in a supporting browser can prevent a site from being rendered inside a frame.
2. 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 hosts line.
3. 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.
4. Cross Site Scripting (XSS) Protection - Django templates protect against the majority of XSS attacks though it has certain limitations with respect to HTML and Database interactions.
5. Session Security – Every session created for the user is tracked and managed in the database for security. This is done using [django.contrib.sessions](https://docs.djangoproject.com/en/3.0/topics/http/sessions/#module-django.contrib.sessions) command.
6. SSL/HTTPS – This feature is provided by Heroku Cloud services hosting the app in a HTTPS server.

A screenshot of a cell phone

Description automatically generated

# Project Development

The development of the

# Testing

# Conclusion

# Proof of Concept

# Reflection

# Bibliography

# Appendix 1

# Appendix 2

*GitHub repository link:*