Portfolio Website

Jevon Lee II

CST-451 Capstone Project Final Architecture & Design

Grand Canyon University

Instructor: Amr Elchouemi

Revision: 3.0

Date: 10/11/22

|  |
| --- |
| History and Signoff Sheet |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Revision Notes** |
|  |  | Initial draft for review/discussion |
|  |  |  |
|  |  |  |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Integrated Instructor Feedback into Project Documentation**

Yes  No

Design Planning Summary

1. One of the biggest issues that face graduates and soon to be graduates of any computer science/programming degree is what to do once graduation is over. Most people wind up using their experiences and projects to set up a resume and submit it to companies and potential employers. But what if instead of coming in with a paper showing your skills and answering all those interview questions that come with the employer not being impressed with you, you were able to walk in and give them a URL to a fully functional and professional website that has everything anyone could be looking for. Yes, it would reduce paper consumption by one, but that is still better than nothing. It is also beneficial to have an online portfolio because it would take the guess work out of demonstrating skills and professionalism as all the employer would have to do is look it up and see for themselves firsthand.

Technical Requirements (Functional & Non-Functional):

* Functional Requirements:
  + Users will be able to see information displayed in formal & professional manner
  + Users will be able to interact with pages using high level development added to pages
  + Admin will be able to use CRUD to show users different types of information relating to Software Development
  + Admin will have access to page settings
  + Admin will be able to login with their own special admin information
* Non-Functional Requirements
  + Effects/Animations for web page should load in a timely manner
  + Web design should be responsive. With the ability to resize to different devices and screen sizes(if desired i.e., half screen, PIP)
  + Site will be stable and not crash along with DB connection being stable and properly loading data
  + Web Design will properly reflect style of creator & not fluctuate between pages, keeping the same themes

Design Concepts Overview:

Below is a picture of the proposed solution of how the flow of the web app is going to be. The site will start by using angular for the front-end work of the app. This will involve incorporating bootstrap and other css methods to enhance the users experience. Springboot is used for the backend of the app. It will incorporate previously learned technologies like Spring Data JDBC and its business service. This will incorporate the user/admin logins as well as data placed into the DB for users viewing. Finally, MongoDB will be implemented for the site DB. In case of emergency or failure to connect to Mongo, MAMP & PhpMyAdmin will be used instead. This is to hold any relative data that will be entered into the site. Microsoft Azure will be used for the cloud service to deploy the web app to, because it is given to students as a free service for GCU students and it is extremely new user friendly.

Diagram

Description automatically generated

User-Interface Screen Concepts(UI):

* Profile Register/Edit Page
  + The profile page is going to show all of the users information from their first registration on the website. Any User will be able to access the full site, but you would need to register in order to send any inquiries or comments on the comments page. Login will be required for further comments. This is also the base of the registration form the user will be presented, with a few changes to buttons and form titles.

Graphical user interface

Description automatically generated

* Main Page/Home Page
  + The main page is where the user will get most of their information regarding the web app creator. Cards will be implemented to demonstrate certain small bits of programming languages learned throughout the program. Each card would be clickable for more information along with a small picture for an example of some sort. The picture near the bottom are previews of the programming page info for users to see before leaving the main page. There will also be a visually pleasing background with an aesthetic that fits the theme of the app.

Graphical user interface

Description automatically generated

* Comments Page/Feedback
  + This is a page created in order to improve on any deficiencies found within the web app. The app should be bug free by the time of deployment, but things can slip through, so any bugs or things users find to be buggy can be reported for fix in future updates. The user can simply leave feedback as well, such as suggestions for theme change, UI fixes and different features that can be added for enhanced user experiences.

Graphical user interface, text, application, website

Description automatically generated

* Demos/Videos Page
  + This page is for any demos that will be added for user experience. When clicking on certain information in the programming tab, it will link to a video for the corresponding item if the user wishes to follow. The demo may be a video from a 3rd party, or a video created by me to demonstrate my knowledge with another source created in a class.

Graphical user interface, application

Description automatically generated

Diagrams:

The project started off by completing the design and concepts for the front-end of the app first. I had begun with brainstorming ideas on what specific pages needed to be implemented in order to fully demonstrate my skills and knowledge learned throughout my time at GCU. From there I was able to work at understand what exactly would go into the backend work, like what specific data am I going to pass through, and certain features of the web app.

* Class Diagram

Diagram

Description automatically generated

Database Schema

A screenshot of a computer

Description automatically generated

Graphical user interface, application, table, Excel

Description automatically generated

Detailed Solution Architecture:

![Diagram

Description automatically generated]()

Registration/Login UML Sequence

Diagram, schematic

Description automatically generated

|  |
| --- |
| Hardware and Software Technologies |
| 1 – VS Code |
| 2 – Spring Boot |
| 3 – MongoDB |
| 4 – Angular |
| 5 – MAMP/ PHP Admin (for testing) |