Degree Explorer

Austin Harvey

CST-451 Capstone Project Proposal

Grand Canyon University

Instructor: Professor Mark Reha

Revision: 1.0

Date: 9/27/2020

**ABSTRACT**

Tech Savvy is a simple and quick way to discover your future career in technology. When a user register, they are prompted to input a job title that will search for jobs in the field of technology. The application has a task of requesting and receiving data to display multiple jobs and their description on a table. The user has the option to view their history searches and save jobs to view in their profile. As an accomplishment, the application is made up of a .Net Core back-end and React front-end that continues to receive data using a REST API from the third-party GitHub Jobs API. Both project ends will transfer data between both ends to be displayed to the user. The secured application has a MYSQL database that hold registered users with security caches and is available on the Azure cloud.

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

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Revision Notes** |
|  |  | 1.0 |
|  |  |  |
|  |  |  |

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

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

**Integrated Instructor Feedback into Project Documentation**

Yes  No

**Project Approval**

Professor Mark Reha

**TABLE OF CONTENTS**

Project Overview and Project Objectives 4

Project Scope 5

Project Success Measures 6

Project High-Level Solution 7

Project Controls 8

Project Cost and Schedule 10

Appendix A – References 11

Appendix B – Copyright Compliance 12

Project Overview and Project Objectives

**State the Problem and Background**

The problems this application will solve are “What sort of technology career will I have and job can I get if I want to pursue this degree?” and “I’m only curious to know what this degree has to offer but where can I go to easily find out?”.

Before starting college, students in technology have a hard time figuring out what they want to do for the rest of their lives, and they are confused on whether that degree will get them their dream job. It is a lot of work to research many job sites to find out what career they might want to do after graduating college. With this application, students will have a better idea on what career they may want to pursue. This application can also help people who are simply wondering what technology jobs are being offered.

**Christian Worldview**

With a Christian worldview perspective, I want this application to help others with their fear of the future and give them some hope on what they want to achieve in their life for Christ. This application should only bring good to the user with an opportunity to follow God’s plan when degree searching.

**Project Objectives**

This project has many objectives to achieve. For one, the application should have two projects for back-end data and front-end display. Next, the back-end project needs to receive data using a REST API and hold it. Data should transfer back-end data to front-end project to be displayed. A search function to retrieves job data with a keyword using REST API. Jobs should display in a nice-looking list. Lastly, a button displays next to each job that redirects user to job application.

**Challenges**

There will be several challenges that will come during this project. Create a working REST API that will request and receive data. Create two projects that will transfer data between each other. Get accurate data to be displayed and look acceptable to the customer. Lastly, create a database that will hold caches and security implementations.

**Benefits and Opportunities**

From implementing the project, bugs and issues alike can be exploited before going live. The design of the front-end project can be reflected upon and be altered to look better. Refactor code to be more agile and secure. The application will be more efficient, and time will be saved through project implementation.

Project Scope

1. In Scope

With a key word, allow registered users to search through a job listing website using a REST API and retrieve certain jobs to be displayed in a list.

Users will have the option to view search history on a separate web page.

Button next to each job to redirect to application website.

1. Out of Scope

Salary per year per job compared by each other on a bar graphs.

Login using third party’s (of API) credentials.

Mobile application.

1. List the work breakdown required to satisfy the project objectives. Identify teams and other resources that may be required to successfully complete the project.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Work Breakdown Structure | | | | | | | | | | |
| ID | Task | Dependencies | Status | Effort Hours | Cost | Start Date | Planned Completion | Estimate to Completion | Actual Completion | Resource |
| 1 | Research | Find out technologies that will be used. | Ongoing | 10 | 0 | Topic 1 | Topic 1 | 100% | Topic 1 | N/A |
| 2 | Develop | Program and test functionality | Inactive | 80 | 0 | Topic 7 | April 2021 | 0% | N/A | N/A |
| 3 | Planning | Plan out tasks and functionality | Ongoing | 10 | 0 | Topic 1 | Topic 8 | 80% | N/A | N/A |

Project Success Measures

1. Describe what measures will be used to calculate project success.

For this project, success is measured mainly by getting a connection between the REST API and the front-end project. The goal is to grab data from a third-party source, display it on the font-end page, and interact with the it.

1. Use the template to list the project completion criteria.

|  |
| --- |
| Project Completion Criteria |
| 1 - Created front-end and back-end projects and connect them to interact with each other |
| 2 – Created a REST API to retrieve data from third-party source |
| 3 – Connected to third-part source and retrieve data |
| 4 – Transferred data from back-end to front-end project |
| 5—Created a database to store caches and user information |
| 6 – Used React to create the front-end page |
| 7– Designed front-end page with bootstrap and JavaScript |
| 8 – Designed a search bar to insert key to retrieved third-party data |
| 9—Displayed third-party data gathered from key word in search bar onto table |
| 10 – Created a button next to each data row on the table to redirect user to application website |
| 11 -- Move application to the cloud |

1. Use the template to list the project assumptions and constraints, if applicable. An assumption is an educated guess that a likely condition or circumstance is presumed to be true. A constraint is a limiting condition or circumstance that defines the project boundaries. Assumptions allow the project to succeed. Constraints restrict or limit the project execution.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Assumptions and Constraints | | | | | |
| ID | Description | Comments | Type | Status | Date Entered |
| 1 | Key word in search word validation | The key word needs to be an existing degree title | Constraint |  | 9/22/2020 |
| 2 | Creating a React front-end project | Never used React before so it will take awhile | Assumption |  | 9/22/2020 |
| 2 | REST API implementation | It will take many hours to create a REST API with headers and bypass security | Assumption |  | 9/22/2020 |

Project High-Level Solution

**Challenge and Objective**

The challenge that is faced for this project is locating and retrieving the correct data from a third-party source using a REST API. Code will need to be created to send and retrieve HTTP requests that will then be transferred from one project to another. Once transferred, the data will be displayed using React. Users will need to register using MYSQL database to interact with the application. Caches will be stored in the database to save previous degree searches and security. The objective is for the registered user to enter a degree title and retrieve a list of jobs to be displayed. The user will have the option to view a second page where they can revisit past searches. Some characteristics of this project solution are how simplistic yet easy to the user it can be to find jobs related to one keyword search. The solution can be very informative and grant insight to the user’s interest and their future occupation. As for assumptions, keywords inputted will need validation to get accurate data using the API. I have no knowledge of React and will need to take more time to learn how to connect the .Net Core project with the React project. Creating a REST API will be the most challenging part of this capstone because it is a process to setup correctly and be secured.

Block Diagram showing the project narrative as a ladder process:

Third-Party API

Gather data from key

Send data

Back-End

Request data

Receive data

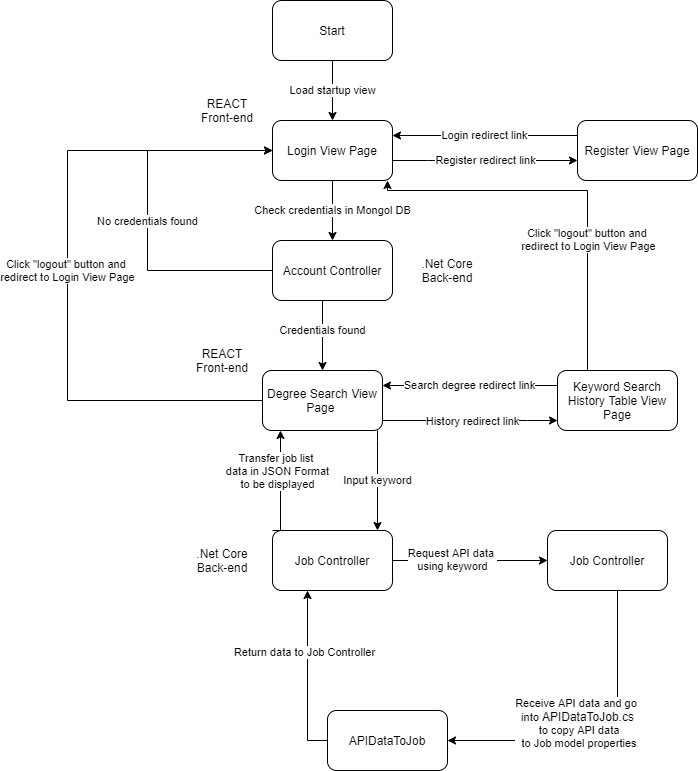
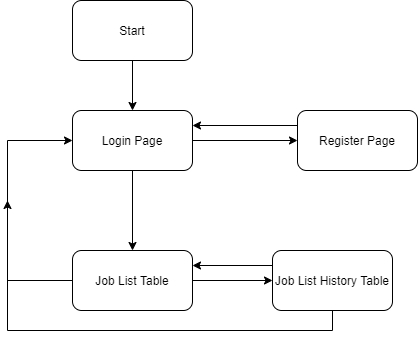
Input key

Display API data

Front-End

**Solution**

The concept of the solution is to remove obstacles from discovering or pursuing a degree. It is important to make the user’s experience better through simplicity and being direct. Confusing the user through either UI navigation or functionality would not benefit the user or the marketing of the application. The solution to this project’s problem is setup as a ladder process by how data is transferred from one step to the next. All steps work one at a time and cannot be skipped. This will allow problems to be identified and resolved quicker and more efficiently. The concept of using a REST API to return accurate job data that the user expects will increase the application popularity. The solution requires a string input to send a HTTP request to the third-party API then post data in JSON format to be displayed in a bootstrap table. This solution will have the approach to complete the objective by allowing the user to request job lists and receive them for the user to interact with.



Front-end Project Sitemap

Project Flowchart

Project Controls

1. Use the template to define the risk and list the steps to prevent the risk from occurring or the steps to minimize the chances of it happening. The contingency plan describes alternative solutions to reduce the impact of the risk. An example of a contingency plan is to provide the customer a temporary web server if there are delays in delivery/completion. If the risk has already happened, then provide an entry in the issue log.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk Management | | | | |
|  | **Risk Probability** | **Risk Impact** |  |  |
| **Event Risk** | **(high, medium, low)** | **Risk Mitigation** | **Contingency Plan** |
| What is the risk? | What is the probability? | What is the impact if the risk occurs? | What can be done to minimize the risk? | What can be done to minimize the impact of the risk? |
| Bypass REST API header security | medium | Collecting data | Researching solutions through tutorials | Find multiple options to bypass header security |
| Connect back-end project to front-end project | medium | Application will not display third-part data | Learn how .Net Core and React can transfer data between each other. | Know how .Net Core and React works |
| A degree title keyword that will return accurate jobs | high | The API may return data that the user did not want | Make sure the API being used can handle keyword search | Have the API to return the correct data. |

Project Cost and Schedule

1. Create a project schedule after all project tasks have been defined and prioritized.

|  |  |
| --- | --- |
| 1. | Propose capstone idea and get accepted |
| 2. | Research technologies to be used and identify risks |
| 3. | Set requirements for project |
| 4. | Design project architecture |
| 5. | Begin developing project |
| 6. | Create back-end and front-end projects (.Net Core and React) |
| 7. | Get access to third-party API (GitHub Jobs API) |
| 8. | Create back end code to call and receive API data |
| 9. | Create front-end view page to display API data on a table |
| 10. | Database to hold users and caches |
| 11. | Login and Register view pages |
| 12. | View page to view previous searched keywords |
| 13. | Add security to the application |
| 14. | Move application to the Cloud (Azure) |
| 15. | Put application on the mobile store (Apple and/or Google Play) |

1. Set a programming schedule by implementing work breakdown and task time estimates. Create a timeline with dates for completion of key components of the project.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Task | Time Estimate | Completion Date (Starting from Development Phase and only days of actual coding) |
| 1 | Create .Net Core project | 5 minutes | Day 1 |
| 2 | Set up basic Model and Controller classes of the MVC design in .Net Core project | 1 hours | Day 1 |
| 3 | Create React Project | 5 minutes | Day 1 |
| 4 | Setup React page with table and get ready to receive data from back-end project using a search bar (Include search validation) | 10 hours | Day 2 - 10 |
| 5 | Setup job search model to hold REST API data in .Net Core | 8 hours | Day 10-13 |
| 6 | Setup Controller REST API to request and receive data from the third-party API | 15 hours | Day 14 - 21 |
| 7 | Bypass REST API header security | 9 hours | Day 22 - 25 |
| 8 | Transfer REST API data from back-end controller to front-end project | 5 hours | Day 26 - 27 |
| 9 | Display REST API data onto bootstrap table in React | 4 hours | Day 28 - 29 |
| 10 | Create and setup MYSQL database to hold users and security caches | 11 hours | Day 30 - 37 |
| 11 | Create React page to view entire keyword search history | 10 hours | Day 38 - 43 |
| 12 | Refactor code and clean front-end HTML and CSS | 6 hours | Day 44 - 47 |

Appendix A – References

*Getting Started –*. (n.d.). React. Retrieved September 26, 2020, from <https://reactjs.org/docs/getting-started.html>

Appendix B – Copyright Compliance

Indeed API License Copyright: <https://www.indeed.com/legal?hl=en&redirect=true#api>

Bootstrap License Copyright: <https://github.com/twbs/bootstrap/blob/v4.0.0/LICENSE>

REACT License Copyright: <https://github.com/facebook/react/blob/master/LICENSE>

.Net Core License Copyright: <https://github.com/dotnet/core/blob/master/LICENSE.TXT>