Table of Contents

1. **PROJECT OVERVIEW**
2. **PROJECT DEFINITION**
   1. Project Description
   2. Project Objective
3. **TEAM MEMBERS AND ROLES**
4. **PROJECT APPROACH AND TECHNOLOGIES**
   1. Project Approach
   2. Project Technologies
   3. Architecture
5. **USER STORIES BACKLOG**
6. **PROJECT SCOPE**
7. **CONSTRAINTS**
8. **ASSUMPTIONS**
9. **PROJECT OUT OF SCOPE**
10. **SPRINTS INITIAL PLANNING**
    1. SPRINT - 1
11. **RISK AND DISCLAIMER**
    1. Risk
    2. Disclaimer

**12. BUDGET AND BUDGET REPORTING**

* 1. Budget Analysis

12.2 Cost Estimations

# Project Overview:

|  |  |  |
| --- | --- | --- |
| **Project Name** | **Project Start Date** | **Project End Date** |
| **Career Service Platform** | **Jul 6,2023** | **August 7,2023** |

1. **PROJECT DESCRIPTION:**

This project aims to provide a comprehensive career services platform where students can showcase their portfolios and employers can post job openings. It offers a convenient and efficient way for job seekers to find relevant employment opportunities and for employers to identify potential candidates. The platform allows students to create and update their profiles, apply for jobs, and track their applications. Employers can create job posts, review candidate applications, and select candidates for interviews. The system prioritizes security, ensuring the protection of login credentials and providing notifications to both employers and students.

### Project Objective:

The project’s goal is to provide job seekers a platform to find and apply for a job and gives an opportunity to the employer to find a potential candidate for various vacant positions in the company. The students/job seeker will get notified for their application status and they should also be able to track their application and schedule their interview with the employer.

# TEAM MEMBERS AND ROLES:

|  |  |  |
| --- | --- | --- |
| **Role** | **Name** | **Description** |
| Full Stack Developer | Chetan Sajjansharan Panchal | Frontend, Deployment and DevOps |
| Quality Analyst and Documentation | Kenish Rajeshbhai Halani | Test Planning, Test Case Design, Defect Reporting and Tracking. |
| Full Stack  Developer | Umang Rakeshbhai Patel | Back-end Development, Database Management |
| Full Stack  Developer | Urvishkumar Kapadiya | Back-end Development, Database Management |
| Full Stack  Developer and Documentation | Dvij Barot | Defect Reporting and Tracking, Test Documentation, Collaboration, and Communication |

1. **PROJECT APPROACH AND TECHNOLOGIES: -**

We will use Agile Project Management with Scrum methodology for our project. Our total project work is divided into short cycles known as “sprints”. Tasks for each sprint is taken from the product backlog for each sprint iteration.

We will use the following development tools to develop the System.

|  |  |
| --- | --- |
| **Tools** | **Name** |
| Product Management | Jira |
| Programming Language | Java |
| Framework | SpringBoot |
| Web Technology | HTML5, CSS3, JavaScript |
| Database | SQL Server or MySQL |
| Version Control | Git |
| IDE | Eclipse |
| Operating System | Windows |
| Application Server | IIS |
| Testing | JMeter |

## Project Approach: -

### Agile and Scrum methodology

Agile is a project management methodology for software development that places a strong emphasis on teamwork, adaptability, and client satisfaction. The Agile methodology is based on the Agile Manifesto, a set of principles for software development that prioritize delivering working software, collaboration, and responding to change. Requirements and solutions in an Agile software development project are developed collaboratively by self-organizing, cross-functional teams. The development process is iterative and incremental, with regular cycles of planning, development, and review.

We can use Agile as follows in developing the project:

1. Prioritize and refine the project backlog: the team works with stakeholders to prioritize the project requirements and refine them into a backlog of specific and actionable items.
2. Plan and estimate: the team breaks down the project backlog into smaller chunks and estimates the time and resources needed to complete each chunk.
3. Develop and deliver: the team delivers small, usable portions of the software (called "iterations" or "sprints") in regular intervals, usually every two to four weeks.
4. Review and retrospect: the team regularly reviews the work completed and holds retrospectives to identify opportunities for improvement and make changes for the next iteration.
5. Collaborate and adapt: Agile encourages close collaboration between the development team, stakeholders, and customers, and values adaptive planning and flexible responses to change.

The Scrum framework can be used to manage the development process and ensure a smooth flow of work from the initial idea to the final implementation. The project will be divided into sprints, with each sprint focusing on delivering a specific set of features and functionality.

The team can prioritize the requirements and user stories based on their importance and urgency and can facilitate daily stand-up meetings, sprint planning and review meetings, and sprint retrospectives to ensure that the project is progressing as planned and that any issues are addressed in a timely manner.

The Agile and Scrum methodology will help to ensure that the project is delivered on time, within budget, and with high-quality results.

## Technologies Used: -

### Version control

Git:

Version control allows us to maintain track of your work and readily investigate the changes we've made, whether it's data, code scripts, notes, or anything else. If we save multiple files with the same name while adding versions in the name by ourselves, then Version control is considerably smoother and easier to apply using version control software such as Git. Using an online site to host our files, such as Github, ensures that we have an online backup of your work, which is advantageous to both us and the client.

### IDE

Eclipse/IntelliJ:

IDE can provide significant time savings, reduce context switching and make coding much easier for you. The use of IDE is necessary for

* + - Debugging: The debugger is a tool for analysing programs on a line-by-line basis, monitoring and altering variables, and watching output as it is generated. This is just not possible with IDEs.
    - Unit Testing: It’s important to do unit testing while doing the development process, this reduces the technical debt by a lot.
    - Code Refactoring: Code refactoring makes it easier to perform global code changes, saving time compared to making changes manually.
    - Source Code Integration: Version Control Systems are also part of IDEs, so no need to push, pull, commit from another cmd tool.
    - Development and Release Tool Integration: Gradle,

A few features that help with coding include spell-checking, track changes and database integrations with the database explorer. For our project, we’ll be using IntelliJ’s eclipse as it’s most useful when dealing with Java based applications.

### OS

Windows:

As the majority of the market is shared by Window devices [ref] and as information provided by client, our main focus will be getting the website to work on Windows systems; other supported versions might be rolled out in the future versions.

[<https://gs.statcounter.com/os-market-share/desktop/worldwide>]

### Product Management

Jira/bitbucket:

The application of specialized knowledge, skills, tools, and processes to produce something of value to others is referred to as project management. Projects include the creation of software to improve a company process, the construction of a building, the relief effort following a natural disaster, and the expansion of sales into a new geographic market. [ref][[https://www.pmi.org/about/learn-about-pmi/what-](https://www.pmi.org/about/learn-about-pmi/what-is-project-management) [is-project-management](https://www.pmi.org/about/learn-about-pmi/what-is-project-management)]

Jira's adaptability and innovative tools have propelled it to the top of project management software rankings throughout the world. Jira is well-known for its Scrum support; it supports Scrum's scaled agile framework, which includes sprint planning, standups, sprints, retrospectives, and scrum teams. To develop an even more customized process, we might select between fixed-length sprints and future sprints, turning on and off agile elements as needed. [ref][[https://producthq.org/agile/product-](https://producthq.org/agile/product-management/jira-for-project-management/) [management/jira-for-project-management/](https://producthq.org/agile/product-management/jira-for-project-management/)]

There’s only one problem, the free version of Jira supports at max 10 people. We acknowledge that, and will try to work around that, as other free alternatives are not as effective as Jira, and not many people from our team have experience with those.

### Application Server:

IIS:

Since the application that is aimed to be built is web-based, Internet Information Services is the best option to use. Request response architecture will be followed by using HTTP protocol as the medium.

### Testing:

* Apache JMeter:

JMeter is an open source Java designed application commonly used for testing functional test cases. JMeter allows efficient coverage of end to end test cases for regression testing of the web application. It also allows use case specific functional tests.

JMeter generates an HTML report that covers all the performance metrics of the test case.

* Jacoco:

Jacoco is a library that generates reports based on the code coverage and can be used as a maven plugin as well. It can be easily integrated with IDEs.

Reference: <https://jmeter.apache.org/>

Reporting engine:

JMeter dashboard:

JMeter supports dashboard report generation to get graphs and statistics from a test plan. The report includes performance metrics including response time, network speed associated with every test label. It also includes an error page listing all the errors occurred and the number of tests in which the error occurred. vGraphical representation of the test cases also covers a lot of these metrics.

Reference for setting the dashboard up: <https://jmeter.apache.org/usermanual/generating-dashboard.html>

### Programming Language:

Java:

The project would be developed in Java which is a high-level and object-oriented programming language. It adheres to the WORA principle of "write once, run everywhere" and is highly scalable.Java is the most secure, open-source and free to use programming language which helps to reduce overall development cost and promotes code reuse.

### Framework:

Maven/SpringBoot:

The Java backend framework will be used to implement the project's backend. The build tools aid in project automation by employing a script to save the source code into machine code.

An open-source project management tool called Maven helps to automate the download of Jar files and other dependencies.By providing a thorough description of the mistake, it maintains repositories and enhances error reporting.

An open-source platform for constructing websites called Spring Boot boosts productivity and cuts down on development time. It has community backing and offers flexibility and scalability, making it the finest technology to be adopted.

### Web Technology

HTML, CSS, JavaScript:

Basically, these are the tools and techniques that are used in creation and maintenance of websites. HTML, CSS and JavaScript are the front end languages used to build the part of websites with which users can interact directly.HTML is used to define the text documentation whereas CSS helps in applying styles to the web pages. JavaScript is used to enhance the website functionality by making it more interactive. The front-end of the project is built using these core web technologies..

### Database:

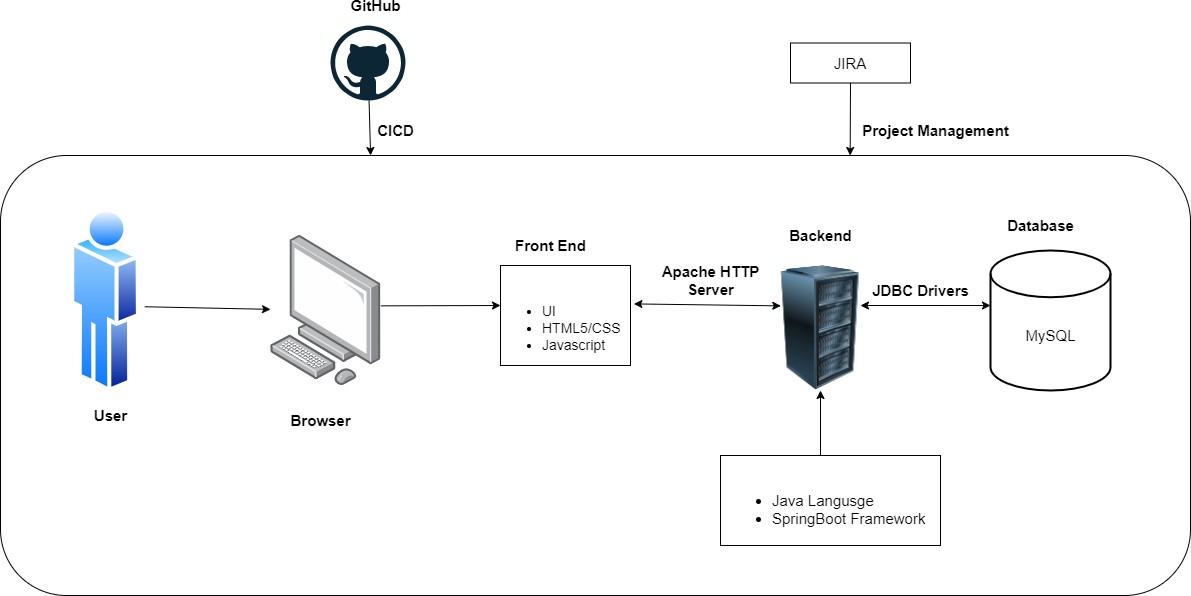
SQL Server:

Data management is the important part of the project and SQL Server is used in this project.

Microsoft SQL Server is relational database software developed by microsoft. It is a collection of tables which consist of rows and columns. It is platform independent and supports case insensitive language named SQL. It maintains production, development and reduces database problems.It also helps in the cost reduction of the software making it the most reliable choice for the projects.

## Architecture: -

For the frontend part of this project, HTML5, CSS3 and javascript will be used and Java programming language will be used for the backend development of this project. In order to implement agile methodology throughout the project JIRA will be used as the project management tool. MySQL will be used to store the key information of patients, doctors, counsellors and managers. For hosting the source code of the project GitHub will be used.



# USER STORIES BACKLOG: -

|  |  |  |
| --- | --- | --- |
| Sr. no | Category | User story |

1. **PROJECT’s SCOPE:**

***Following Functionalities will be developed for the web-based application:***

1. **Registration and Login Modules**

In this module all the users

1. **Students/Job seeker Module**

This module will help the

1. **Reports Module for Students**

Once the.

1. **Notification/Alert Modules.**

This module will send the notification and alerts of the events and activities to the concerned user.

1. **Interview Management Module.**

After examining the results

1. **Management modules for**

This module will keep the data of all the students, and employees that have accessed the system and can view, edit, and update their details.

1. **Dashboard for all the users.**

A dashboard for all the users so that they can have a glimpse of all the activities going on.

1. **Access Rights and role management module.**

This module will manage all the roles and grants access to user based on the role. Admin can change the access rights of all the users.

# 7. ASSUMPTIONS: -

1. All users are assumed to be familiar with GUI terminals.
2. All users have an overview of the user training.
3. As the deadline of the project is tight all the team members might work on multiple roles and put in extra efforts to deliver user stories on time.
4. No integration of third-party apps.

# 8. CONSTRAINTS: -

1. Time Constraint- The project has a tight deadline; the project is planned into 3 sprints of 1 weeks each.
2. Quality Constraint- The final application might not meet /satisfy all the user requirements because of tight budget and limited time.
3. Resources Constraint- Availability of all the resources and technologies is very important for the successful implementation of the project.

# 9. PROJECT OUT OF SCOPE

1. Two Factor Authentication
2. Adding Multiple profiles
3. Mobile Application for this project
4. Verification through One time password (OTP)
5. All the functionalities that are not mentioned in project scope are by default considered out of scope.

# SPRINTS INITIAL PLANNING

#### Sprint1: -

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr No | Sub Task | Epic | User Story ID | Story Points | Description |
| 1 | Remote repository creation | Repository setup |  | 2 | * Create folder structure of the repository * Create the remote repository in GitHub * Provide access to the team members |
| 2 | Database Design | Database setup |  | 2 | * Draw ER diagrams and design diagrams to finalise the tables * Finalise on the attributes to be present in each table * Identify the primary keys and the foreign key constraints between the tables |
| 3 | Database creation |  | 2 | * Create the tables in a MySQL database with respective attributes and dependencies. * Export the .db file and update in the remote repo |
| 4 | Build the initial documentation | Registration and Login component | 1 | 2 | * Documentation about the initial phase of the project where the roles and the responsibilities of the team will be decided. |
| 5 | Build the basic Functionality (Student) | 2 | 2 |  |
| 6 | Build the Registration form | 8 | 1 |  |
| 7 | Build the login for admin | 9 | 1 |  |

1. **Risk and Disclaimer**

# Disclaimer

# “write about disclaimer”

Risk Management and Mitigation Techniques-

We identified the following risks with the delivery of this project: People risk: • Front end - While discussing our previous software engineering experience, we came to know that most of us have little front-end experience. Avoidance - We chose to use simple HTML, CSS and Javascript with Bootstrap instead of using frameworks like React and Angular, which we were initially discussing, as the time spent in learning these may affect our deliverables. • Back end - We identified that although we do have team members with back-end experience, they were in different technologies (Springboot).

**Mitigation** - We chose to use Django to mitigate this risk as this framework works well with Java, and we came to the conclusion that this is easier to learn when it comes to implementing a project of this size, compared to the heavily structured approach using Java-Spring would have required.

Organizational risk:

* 1. There is a possibility that any one of our eight team members could drop the course at any time.

**Mitigation** - The role of the project manager is handled by two team members, and we plan to not have developers work on features in silos, so as to have a good understanding of the codebase by most developers and minimize dependencies.

Estimation risk:

* 1. The time required to implement the story features might be underestimated.

**Mitigation** - We used multiple rounds of the Planning Poker technique to assign Story Points, until all of the team members were one the same page. The primary benefit of this was that it allowed everyone to put forward the challenges we might face during implementation and estimate accordingly.

Risk of underestimation would require 1 extra Sprint of effort.

Hourly cost of team (5 \* $40 Dev) + (2 \* $50 PMs) + (1 \* $45 Dev/UX)

= $345

|  |  |
| --- | --- |
| Cost of Planning Poker | 2 hours \* $345 = $690 |
| Cost of 1 extra Sprint | 7 hours \* 2 weeks \* $345 = $4,830 |
| Probability before | 0.3 |
| Probability after | 0.1 |
| Risk Exposure (RE) before | $4,830 \* 0.3 = $1,449 |
| Risk Exposure (RE) after | $4,830 \* 0.1 = $483 |
| Risk Reduction Leverage (RRL) | (1449 - 483) / 690 = 1.4 |

1. **BUDGET AND BUDGET REPORTING**

* Base Costs

|  |  |
| --- | --- |
| Time spent per person per day | 1 hour |
| Total time of all team members per week | 56 hours |
| Total time per Sprint | 112 hours |
| Total time for Project (3 Sprints) | 336 hours |
| Average cost of 1 Person-hour | $345 / 8 = $43.125 |

|  |  |
| --- | --- |
| Cost of Total Person-hours | $14,490 |

* Additional Costs

|  |  |
| --- | --- |
| 6 initial planning meetings | 6 hours \* 8 team members = $2070 |
| Planning Poker | 2 hours \* 8 team members = $690 |
| Contingency Risk Cost | 0.1 Probability \* $4830 = $483 |
| Contingency Integration Factor | 5% of $14,490 = $724.5 |

* Total Cost

|  |  |
| --- | --- |
| Base Cost | $14,490 |
| Additional Costs | $3,967.5 |
| Total Cost | $18,457.5 |