Software Requirements Specification

for

Students Guide

Version 1.0 approved

Prepared by

Labheshwar (Group Lead, MM23)

IMCS, UoS

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to outline the requirements and specifications for the development of a Progressive Web App-based student portal. The portal aims to provide a platform for university students, teachers, and passed out students to connect, share knowledge and resources, and provide guidance to each other.

Intended Audience and Reading Suggestions

This Project is for all those who are searching for parking space and for those who've parking area but of no use for themselves. This Project could be beneficial for both parking area provider and the purchaser/customer who is searching for it.

1.2 Intended Audience and Reading Suggestions

This project is for Educational Institutes, such as Universities/Colleges. This includes Students, Teachers and Admins who are seeking a platform for managing and delivering educational or training content and helping out others.

1.3 Scope

The web-based application will be developed as a progressive web app to ensure it is accessible to all students on all devices. The platform will be designed to allow students to connect with senior students, access recommended courses from Google/YouTube based on their skills and interests, and connect with their teachers for guidance. The platform will also feature frequently asked questions (FAQs) about the university and a 4-year guide to help students avoid confusion and failure. Additionally, students doing jobs in software houses and teachers can post internships/jobs for needy individuals.

1.4 References

We were searching for important tools and libraries related to our project and we came to some important resources.

- MERN Stack Guide: https://www.mongodb.com/mern-stack
- React documentation: https://reactjs.org/docs/getting-started.html
- Node.js documentation: https://nodejs.org/en/docs/
- MySQL

- Socket.io: Socket.IO
- Recommendation System: Build a Content-based recommendation engine in JS - DEV Community
- For University Registration: Hipo/university-domains-list: University Domains and Names Data List & API (github.com)
- Sequelize: sequelize - npm (npmjs.com)

2. Overall Description

2.1 Product Perspective

- Target Market: The platform targets students from all academic levels, including undergraduate, graduate, and postgraduate students, who are looking for academic and career guidance. It also aims to cater to the needs of students who are currently doing jobs in software houses or other organizations.

- Platform Features: The platform will have a user-friendly interface that will allow students to easily navigate through its different sections. They will be able to search for senior students and teachers who can mentor them and guide them through their academic and professional journey. The recommended courses section will provide personalized recommendations to students based on their interests and skills. The platform's FAQ section will answer common questions about university life and academic requirements, and the 4-year guide will help students plan their academic path to success. Students and teachers will also be able to post job and internship opportunities for needy individuals.

- Hardware and Software Requirements: The platform will be accessible via any internet-enabled device, including desktops, laptops, tablets, and smartphones. It will be developed using web-based technologies such as Node + Express and React + MongoDB/JavaScript Tech Stack, and other web frameworks. The platform will require a stable internet connection to function properly.

- User Interface and User Experience Design: The platform's user interface will be designed to be simple, intuitive, and visually appealing. The user experience will be seamless and efficient, with easy-to-understand navigation menus, and quick access to the platform's key features. The design will also ensure that the platform is accessible and user-friendly for students with disabilities.

- Expected Benefits: The platform's primary objective is to help students succeed in their academic and professional pursuits. By providing guidance, resources, and job/internship opportunities, the platform can increase student retention rates and improve their academic and career outcomes. The platform can also help the university by improving its reputation as a supportive and inclusive institution that prioritizes the success of its students.

2.2 Product Functions

- **Account creation and login:** The platform should allow students to create an account using their university credentials or other login methods. This will enable them to access the platform's features, connect with mentors and teachers, and apply for job/internship opportunities.

- **Search function:** The platform should provide a search function for students to find senior students and teachers who can mentor and guide them. The search function should be intuitive and easy to use, allowing students to filter and sort results based on different criteria such as academic background or career interests.

- **Personalized course recommendations:** The platform should provide personalized course recommendations to students based on their skills and interests. The recommendations should be generated using machine learning algorithms and data from Google/YouTube, ensuring that students receive the most relevant and up-to-date information.

- **Job and internship postings:** The platform should allow students and teachers to post job and internship opportunities for needy individuals. This will enable students to find relevant work experience and gain valuable skills that can enhance their academic and professional prospects.

- **FAQ section:** The platform should provide a comprehensive FAQ section that addresses common questions about university life and academic requirements. This will help students avoid confusion and frustration, and enable them to make informed decisions about their academic and career paths.

- **4-year guide:** The platform should provide a 4-year guide to help students plan their academic path to success. The guide should include information on course requirements, major/minor options, study abroad opportunities, and career development resources.

- **User-friendly interface:** The platform should have a user-friendly interface that is easy to navigate, with clear menus and buttons. The interface should be visually appealing and accessible to all users, regardless of their technical skills or abilities.

- **Responsive design:** The platform should have a responsive design that adapts to different screen sizes and resolutions, ensuring that it is accessible on all devices, including desktops, laptops, tablets, and smartphones.

- **Secure and reliable:** The platform should be secure and protect user data with appropriate encryption and authentication mechanisms. It should also be reliable and available 24/7 with minimal downtime for maintenance or updates, ensuring that students can access its features whenever they need them.

2.3 User Classes and Characteristics

In this system, there would be 4 user classes that would be performing their own activities as per role:

- **Student:**

  Characteristics: The student is a user of the platform who has registered with their basic and educational information. They can create posts, comment, like and chat with everyone, access recommended courses, view university information, and make public posts for all universities. They can also access the forum page for job/internship postings and queries asking, and can post attachments/images.

  Needs: Students may need access to personalized resources, such as recommended courses and mentorship opportunities based on their interests and qualifications. They may also need a seamless user experience for posting, commenting, liking, and chatting, with easy navigation, clear instructions, and responsive support.

- **Teacher:**

  Characteristics: The teacher is a user of the platform who can perform all the functions of a student, plus accept and reject student post approval requests, announce notifications on the notice board page for their own university, and report a specific user/post that should be shown to admin.

  Needs: Teachers may need access to certain system features and data, such as student posts and activity logs, to perform their duties. They may also need to ensure that the platform content is relevant to their courses and students.

- **Admin:**

  Characteristics: The admin is a user of the platform who can register themselves by domain email of the university, add/remove teachers/students/and other admins, and make any teacher an admin. They are basically a university login middleware.

  Needs: Admins may need to have access to all system features and data, including user information, activity logs, and performance metrics. They may also need to ensure that the platform complies with university policies and regulations.

- **Superadmin:**

  Characteristics: The superadmin is the highest level of access and authority on the platform, responsible for managing and maintaining the system. They can perform CRUD on all universities, teachers, students, admins, and other system resources.

  Needs: The superadmin may need to have access to all system features and data, including user information, activity logs, and performance metrics. They may also need to ensure that the platform complies with university policies and regulations

2.4 Operating Environment

- Web Browser: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari
- Operating system: Windows, Linux, macOS, iOS, Android
- Database: MongoDB
- Platform: React.js for Frontend, Node.js for Backend

2.5 Hardware Interfaces

- No specific hardware requirements for the web application
- Accessible on any device with a web browser and internet connection
- Optimal performance and user experience recommended with a stable internet connection and modern web browser.

2.6 Software Interfaces

- VS Code
- MongoDB Atlas
- NPM
- Browser
- JSON
- RESTful APIs
- Other 3rd Party Services

3. Functional Requirements

- The platform should allow students to create an account and log in using their university credentials.
- The platform should provide a search function for students to find senior students and teachers.
- The platform should provide personalized course recommendations to students based on their skills and interests.
- The platform should allow students and teachers to post job and internship opportunities for needy individuals.
- The platform should provide a comprehensive FAQ section that addresses common questions about university life and academic requirements.
- The platform should provide a 4-year guide to help students plan their academic path to success.

4. Other Nonfunctional Requirements

- The platform should be accessible on all devices, including desktops, laptops, tablets, and smartphones.

- ➢ The platform should be developed as a progressive web app to ensure fast loading times and a seamless user experience.
- ➢ The platform should have a responsive design that adapts to different screen sizes and resolutions.
- ➢ The platform should be designed to be user-friendly and easy to navigate, with clear menus and buttons.
- ➢ The platform should be secure and protect user data with appropriate encryption and authentication mechanisms

Project Schema

```
// Define the schema for a user
const UserSchema = new Schema({
  name: String,
  email: String,
  password: String,
  role: {
    type: String,
    enum: ['Student', 'Teacher', 'Admin', 'Superadmin'],
    default: 'Student'
  },
  university: {
    type: ObjectId,
    ref: 'University'
  },
  bio: String,
  education: [{
    institution: String,
    degree: String,
    fieldOfStudy: String,
    from: Date,
    to: Date,
    grade: String,
    description: String
  }],
  graduation: Date,
  interests: [String],
  skills: [String],
  createdAt: {
    type: Date,
    default: Date.now
  }
});

// Define the schema for a university
const UniversitySchema = new Schema({
  name: String,
  domainEmail: String,
  admins: [{
    type: ObjectId,
    ref: 'User'
  }],
  teachers: [{
    type: ObjectId,
    ref: 'User'
  }],
  students: [{
    type: ObjectId,
    ref: 'User'
  }],
  createdAt: {
    type: Date,
    default: Date.now
  }
});

// Define the schema for a post
const PostSchema = new Schema({
  title: String,
  content: String,
  author: {
    type: ObjectId,
    ref: 'User'
  },
  university: {
    type: ObjectId,
    ref: 'University'
  },
  isPublic: Boolean,
  likes: [{
    type: ObjectId,
    ref: 'User'
  }],
  comments: [{
    author: {
      type: ObjectId,
      ref: 'User'
    },
    content: String,
    createdAt: {
      type: Date,
      default: Date.now
    }
  }],
  createdAt: {
    type: Date,
    default: Date.now
  }
});

// Define the schema for a notification
const NotificationSchema = new Schema({
  content: String,
  university: {
    type: ObjectId,
    ref: 'University'
  },
  author: {
    type: ObjectId,
    ref: 'User'
  },
  createdAt: {
    type: Date,
    default: Date.now
  }
});

// Define the schema for a job/internship
const JobSchema = new Schema({
  title: String,
  content: String,
  author: {
    type: ObjectId,
    ref: 'User'
  },
  university: {
    type: ObjectId,
    ref: 'University'
  },
  isPublic: Boolean,
  attachments: [String],
  createdAt: {
    type: Date,
    default: Date.now
  }
});

// Define the schema for a query
const QuerySchema = new Schema({
  title: String,
  content: String,
  author: {
    type: ObjectId,
    ref: 'User'
  },
  university: {
    type: ObjectId,
    ref: 'University'
  },
  isPublic: Boolean,
  comments: [{
    author: {
      type: ObjectId,
      ref: 'User'
    },
    content: String,
    createdAt: {
      type: Date,
      default: Date.now
    }
  }],
  attachments: [String],
  createdAt: {
    type: Date,
    default: Date.now
  }
});

// Define the schema for a chat message
const ChatMessageSchema = new Schema({
  content: String,
  sender: {
    type: ObjectId,
    ref: 'User'
  },
  receiver: {
    type: ObjectId,
    ref: ‘User’
}
});
```

THANKS