A MINI PROJECT REPORT ON

STUDENT SKILL PROGRESS AND MANAGEMENT SYSTEM

A dissertation submitted in partial fulfilment of the

Requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

Submitted by

Cheruku Bhavana (20B81A1269)

Kandari Deekshitha (20B81A1271)

Gattupally Sreshta (20B81A12B0)

Under the esteemed guidance of

Mr. A Seetharam Nagesh

Sr.Assistant Professor, IT Department

CVR College of Engineering

DEPARTMENT OF INFORMATION TECHNOLOGY

CVR COLLEGE OF ENGINEERING

ACCREDITED BY NBA, AICTE & Affiliated to JNTU-H

Vastunagar, Mangalpally (V), Ibrahimpatnam (M), R.R. District, PIN-501 510

2023-2024

Cherabuddi Education Society's
CVR COLLEGE OF ENGINEERING
(An Autonomous Institution)
ACCREDITED BY NATIONAL BOARD OF ACCREDITATION, AICTE
(Approved by AICTE & Govt. of Telangana and Affiliated to JNT University)
Vastunagar, Mangalpalli (V), Ibrahimpatan (M), R.R. District, PIN - 501 510
Web : http://cvr.ac.in, email : info@cvr.ac.in
Ph : 08414 - 252222, 252369, Office Telefax : 252396, Principal : 252396 (O)

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the Project Report entitled **"Student Skill Prgress and Management System"** is a bonafide work done and submitted by **Cheruku Bhavana (20B81A1269)**, **Kandari Deekshitha (20B81A1271)**, **Gattupally Sreshta (20B81A12B0)** during the academic year 2023-2024, in partial fulfilment of requirement for the award of Bachelor of Technology degree in Information Technology from Jawaharlal Nehru Technological University Hyderabad, is a bonafide record of work carried out by them under my guidance and supervision.

Certified further that to my best of the knowledge, the work in this dissertation has not been submitted to any other institution for the award of any degree or diploma.

INTERNAL GUIDE

Mr. A Seetharam Nagesh

Sr.Assistant Professor, IT Department

HEAD OF THE DEPARTMENT

Dr. Bipin Bihari Jayasingh

Professor, IT Department

PROJECT COORDINATOR

Mrs. G.Sunitha Rekha

Sr.Assistant Professor, IT Department

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The satisfaction of completing this project would be incomplete without mentioning our gratitude towards all the people who have supported us. Constant guidance and encouragement have been instrumental in the completion of this project.

First and Foremost, We thank the Chairman, Principal, Vice Principal for availing infrastructural facilities to complete the major project in time.

We offer our sincere gratitude to our internal guide **Mr. A Seetharam Nagesh,** Sr. Assistant Professor, IT Department, CVR College of Engineering for his immense support, timely co-operation and valuable advice throughout the course of our project work.

We would like to thank the Professor In-Charge of Projects, **Dr. J. Sengathir**, Professor, Information Technology for his valuable suggestions in implementing the project.

We would like to thank the Head of Department, Professor **Dr. Bipin Bihari Jayasingh**, for his meticulous care and cooperation throughout the project work.

We are thankful to **Mrs. G. Sunitha Rekha,** Project Coordinator, Sr. Assistant Professor, IT Department, CVR College of Engineering for her supportive guidelines and for having provided the necessary help for carrying forward this project without any obstacles and hindrances.

We also thank the **Project Review Committee Members** for their valuable suggestions.

DECLARATION

We hereby declare that the project report entitled "**Student Skill Progress and Management System**" is an original work done and submitted to IT Department, CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad in partial fulfilment of the requirement for the award of Bachelor of Technology in **Information Technology** and it is a record of bonafide project work carried out by us under the guidance of **Mr. A Seetharam Nagesh, Sr. Assistant Professor, Department of Information Technology.**

We further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other Institute or University.

_____

Signature of the Student

(Cheruku Bhavana)

(20B81A1269)

_____

Signature of the Student

(Kandari Deekshitha)

(20B81A1271)

_____

Signature of the Student

(Gattupally Sreshta)

(20B81A12B0)

ABSTRACT

In today's fast-paced world, students face many challenges when it comes to managing their academic and professional development. With a plethora of resources available online, it can be overwhelming for students to navigate through the various platforms and tools to find the resources they need to achieve their goals. Additionally, students often struggle with creating an effective resume, managing their career goals, and keeping track of their learning progress. The Student Skill Progress and Management System is a comprehensive web-based application built using REACTJS that aims to provide students with a holistic platform for managing their skills, career, and learning resources.

The system integrates multiple modules, including a dynamic interactive resume builder, learning management system (LMS) tools to manage students' projects and study schedules, career management tools, and accessories modules, to create a one-stop solution for students to effectively manage their academic and professional development.

This application uses REACTJS (A JavaScript library) coding as frontend, for building a single page application, and Spring Boot as a backend framework. The firebase database maintains the details of students enabling them to access their data anytime, anywhere from any device. The use of REACTJS ensures the development of a dynamic, interactive, and high-performance web application with reusable components, effiecient rendering and easy state management contributing to an enhanced user experience. This project thus helps students do their works quicker and in a planned manner over traditional methods, which reduces time and efforts to plan and complete work.

LIST OF FIGURES

TABLE OF CONTENTS

1. INTRODUCTION

As building graduates get ready to enter the workforce, they should explore an inexorably intricate procedure of realizing where to search for a vocation, what to look like, and how to get employed. Moreover, 40% of science fouryear certification beneficiaries leave designing inside three years of graduation. So as to manufacture a differing and exceedingly talented workforce, explore is expected to all the more likely handle the understudies' basic leadership process and the environment of assets and backings they draw upon as they look for their first post-undergrad work.

To address these concerns, the Student Information Management And Tech Solution (SIMATS) can be used by students in maintaining the progress and information management. It helps in building steps for their career development. This app uses React.js coding, which helps in reuse of code various platforms (such as Web Apps and Hybrid Apps). React JS basically is an Open Source JavaScript library which is used for building UI specifically for single page applications. It’s used for handling view layer for web and mobile applications. React also allows to create reusable User Interface components.

React library was first created by Jordan Walke, software developer at Facebook. React allows developers to create large web-based apps which can change data without reloading the page. It works only on UI in application. The main purpose of ReactJS is to be fast, simple, and scalable. This corresponds to view in the MVC (Model–view–controller) template. It can also be used with other frameworks or JavaScript libraries, such as Angular JS. We can render React on the server-side. Easy to know how a component is rendered, just look at the render function. JSX makes it easy to read the code of your components. It is also really easy to see the layout, or how components are plugged/combined with each other. it is easy to test, and can also integrate some tools like jest. it ensures readability and makes maintainability easier. We can use React with any framework as it is only a view layer.

1.1 Literature Survey

Andreas Papadakis, et al.,: "Profiling Students' Performance and Measuring their Progress in the area of Multimedia Communications"

In this paper, a system for profiling students' progress in engineering disciplines is presented and built. The framework is based on current engineering simulation tools that are commonly used in higher education.

Anand Desai, et al.,: "Student Profiling to improve teaching and learning: A Data Mining Approach"

This study focuses on how data mining can be used in an educational environment. The paper tries to classify pupils based on a variety of factors, including their academic records, such as marks in 10^{th} and 12^{th} grades, "an aptitude test, a grit test, and the CGPA score".

1.2 Need for Student Skill Progress and Management System

The need for a Student Skill Progress and Management System (SSPMS) arises from several key factors in modern education and skill development:

Career Preparation: Students can build and showcase interactive resumes, demonstrating their skills to potential employers, improving their job prospects.

Engagement and Motivation: Interactive features and progress tracking in an SSPMS can motivate students to actively participate in their learning journey.

Evolving Educational Landscape: The demand for continuous learning, online courses, and professional development has grown. An SSPMS meets the needs of both traditional educational institutions and lifelong learners.

Competitive Advantage: Institutions that offer a modern, technology-driven SSPMS can attract more students and stay competitive in the educational market.

Data Security and Privacy: An SSPMS can handle student data securely.

2. SOFTWARE REQUIREMENT SPECIFICATIONS

2.1 Functional Requirements:

Dynamic Interactive Resume Builder Module

The Dynamic Interactive Resume Builder is a tool designed to help students create and customize their resumes. It goes beyond a traditional static resume template by providing dynamic and interactive features that enhance the resume-building process and make the resumes more engaging and informative.

Dynamic Content:

The module allows students to easily add and organize different sections in their resumes, such as personal information, objective statement, education, work experience, skills, projects, certifications, and more. Students can dynamically reorder these sections based on relevance.

Learning Management System (LMS) Tools Module

The LMS Tools Module is designed to track student progress and provide performance analytics in a learning environment. This module is essential for educational institutions and online courses to monitor student achievements, assess learning outcomes, and optimize the learning experience.

Course Management: Students can keep track of their learning courses and exam schedules.

2.2 Non-Functional Requirements:

Usability: User-friendly interface that is easy to navigate.

Usability refers to how easily and efficiently users can interact with the system. A user-friendly interface is crucial to ensure that users, in this case, students, can intuitively navigate and use the Dynamic Interactive Resume Builder and Learning Management System (LMS) Tools Module. Some key aspects are:

Intuitive Design: The interface should have clear and logical navigation, making it easy for students to find and use the features they need. Menus, buttons, and links should be well-organized and labeled appropriately.

Consistency: Maintain a consistent design throughout the system. This includes consistent layout, terminology, and visual elements. Consistency reduces confusion and improves user understanding.

Security: Uses encryption to ensure data privacy and confidentiality.

Security is paramount, especially when dealing with personal data and educational records. Encryption plays a crucial role in protecting sensitive information. The passwords of students are to be encrypted properly.

Compatibility: Responsive and adapt to different screen sizes and resolutions.

Compatibility ensures that the system works effectively across various devices and screen sizes, providing a seamless experience for all users. Compatibility is achieved by:

Responsive Design: Design the system to be responsive, meaning it should automatically adapt its layout and content based on the user's device, whether it's a desktop, tablet, or smartphone.

Device Compatibility: Test the system on different devices with varying screen sizes and resolutions. This ensures that users can access and use the system regardless of the device they prefer.

2.3 System Specifications

Hardware Requirements

Processor: Intel Core i5 or higher.

RAM: 8GB or more.

Storage: 500 GB HDD or higher.

Internet Connectivity : Broadband internet connection.

Software Requirements

Operating System : Windows 10, macOS or Linux

Web Browser : Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari

Node.js : Version 10.16.0 or higher.

NPM : Version 6.9.0 or higher.

ReactJS : Version 16.8.6 or higher.

Firebase : Version 9.

Express.js : Version 4.16.0 or higher.

Git : Version 2.17.0 or higher.

Visual Studio Code.

3. DESIGN

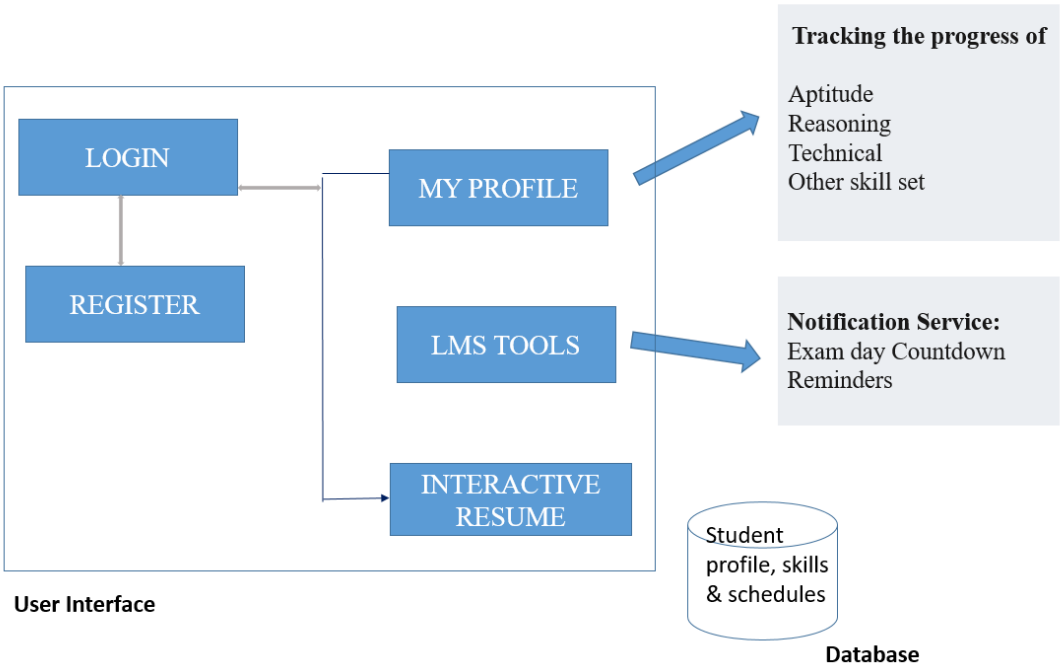

Fig.3.1 *System Architecture*

The system architecture of the Student Management Project is a crucial aspect that outlines how various components and modules interact to create a functional and scalable platform. The architecture ensures that the project is efficient, maintainable, and capable of handling the expected user load.

Client-Side:

Web Browser: Users interact with the platform through web browsers (e.g., Chrome, Firefox, Safari, Edge). The front-end components are primarily built using ReactJS, ensuring a dynamic and responsive user interface.

Front-End:

Front-End Framework: The front-end is developed using ReactJS, providing a component-based architecture that promotes reusability and modularity. React allows for dynamic rendering of content and a smooth user experience.

User Interface (UI): The UI is responsible for presenting information, handling user interactions, and displaying data from the server. It incorporates modern design principles for a visually appealing and user-friendly experience.

Authentication: The front-end handles user authentication, securely managing login, registration, and user profile updates. It communicates with the back-end to validate user credentials.

Interactive Resume Builder: This component allows students to create, customize, andpreview resumes. It utilizes dynamic templates and real-time previews, providing an engagingresume-building experience.

LMS Tools Interface: The front-end interacts with the Learning Management System (LMS) Tools module, enabling students to enroll in courses, access course materials, and track their progress.

Back-End:

Database: The system uses a Firebase to store user profiles, course information, resume data, and other relevant data. It is also used for secure user authentication.

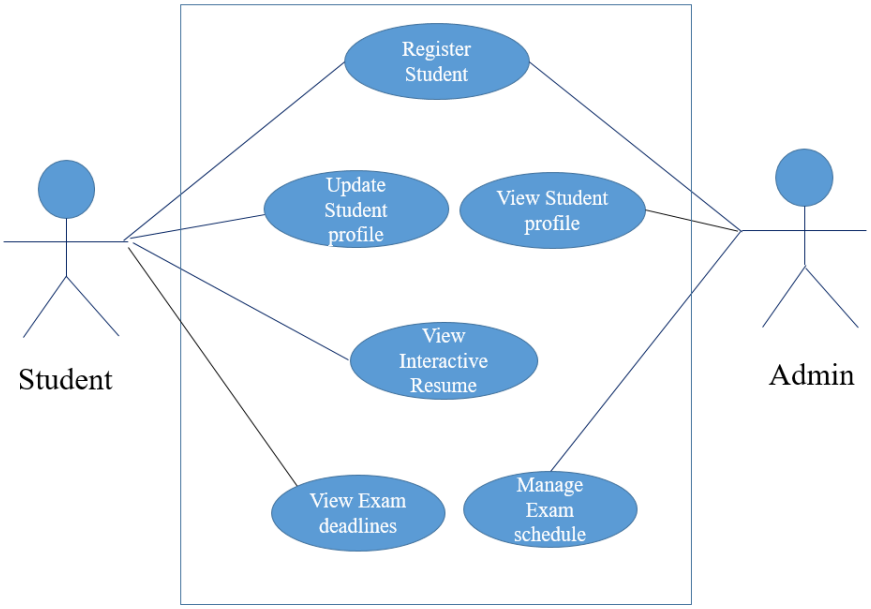

Fig.3.2 *Usecase Diagram*

Student:

***Description*:** The "Student" is a primary user of the system, representing individuals who enroll in courses, track their skill progress, and utilize the available tools for learning enhancement.

Roles and Responsibilities:

Register: Students can register in the system by providing necessary personal information and creating an account.

Update Profile: Students have the ability to update their profiles, including personal details, preferences, and profile picture.

View Progress: Students can track their progress within enrolled courses, viewing completed modules, assignment scores, and overall course progress.

Build Interactive Resume: Students can create and customize their resumes using the interactive resume builder, showcasing their skills and experiences.

View Exam Deadlines: Students can access information about upcoming exam deadlines, helping them plan their studies effectively.

View Interactive Resumes: Admins can view interactive resumes created by students, gaining insights into their skills and experiences.

Both actors play crucial roles in the successful functioning of the "Student Skill Progress and Management System," with distinct responsibilities that contribute to an effective learning and management experience.

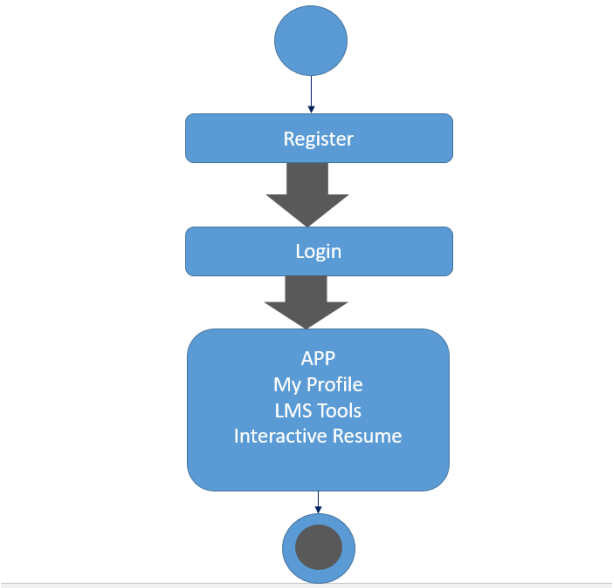

Fig.3.3 *Activity Diagram*

Description of Activities:

User Actions: This is the starting point where users choose the action they want to perform. They can select between registration or login.

***Register*:** Users initiate the registration process, providing necessary information to create a new account within the system.

***Login*:** Users log in with their credentials, verifying their identity and gaining access to the system.

Interact with System: This activity represents the combined interaction of the user within the system, encompassing different functionalities within "My App," "My Profile," "LMS Tools," and "Interactive Resume."

***My App*:** Users interact with the core features and functionality of the application.

***My Profile*:** Users access and update their profile information.

***LMS Tools*:** Users engage with the Learning Management System, including course management, progress tracking, and more.

Interactive Resume: Users access and customize their resumes using the interactive resume builder.

This activity diagram emphasizes the sequence of actions: registration, login, and the combined interaction with various system functionalities. It provides an overview of the user's engagement with the application, showcasing the main activities that users can perform within the "Student Skill Progress and Management System."

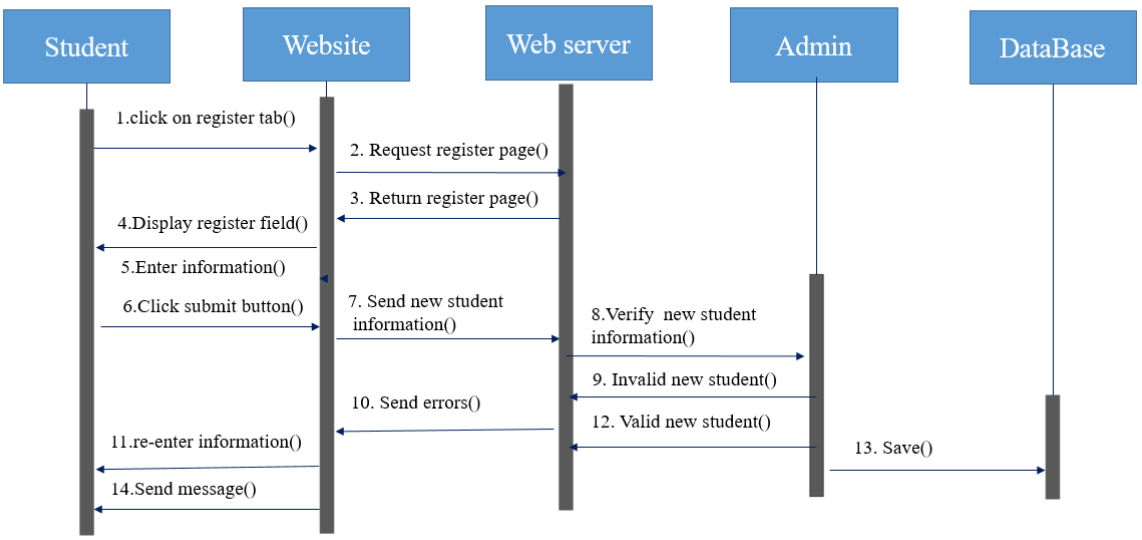

Fig.3.4 *Sequence Diagram*

This comprehensive sequence diagram showcases the multi-step interactions within the "Student Skill Progress and Management System." It commences with the "Student" entity's registration request, followed by a login attempt, which validates the student's credentials. Upon successful login, the student interacts with the "Website," which in turn communicates with the underlying "Web Server." Through this interaction, the student engages in diverse actions, including viewing available courses, enrolling in selected courses, accessing course materials, submitting assignments, and tracking their overall progress.

Simultaneously, the "Admin" entity interfaces with the "Web Server," enabling a range of administrative actions such as managing courses, controlling student enrollment, uploading course materials, grading assignments, and monitoring students' progress in courses. All these interactions occur within the context of a centralized "Database," ensuring data consistency and accuracy throughout the system.

The sequence diagram highlights the dynamic flow of actions, interactions, and data exchanges, illustrating the coordinated functioning of students, administrators, the website, the web server, and the database. This orchestration supports effective course management, skill development tracking, and seamless user experiences, ultimately enhancing the overall functionality of the "Student Skill Progress and Management System."

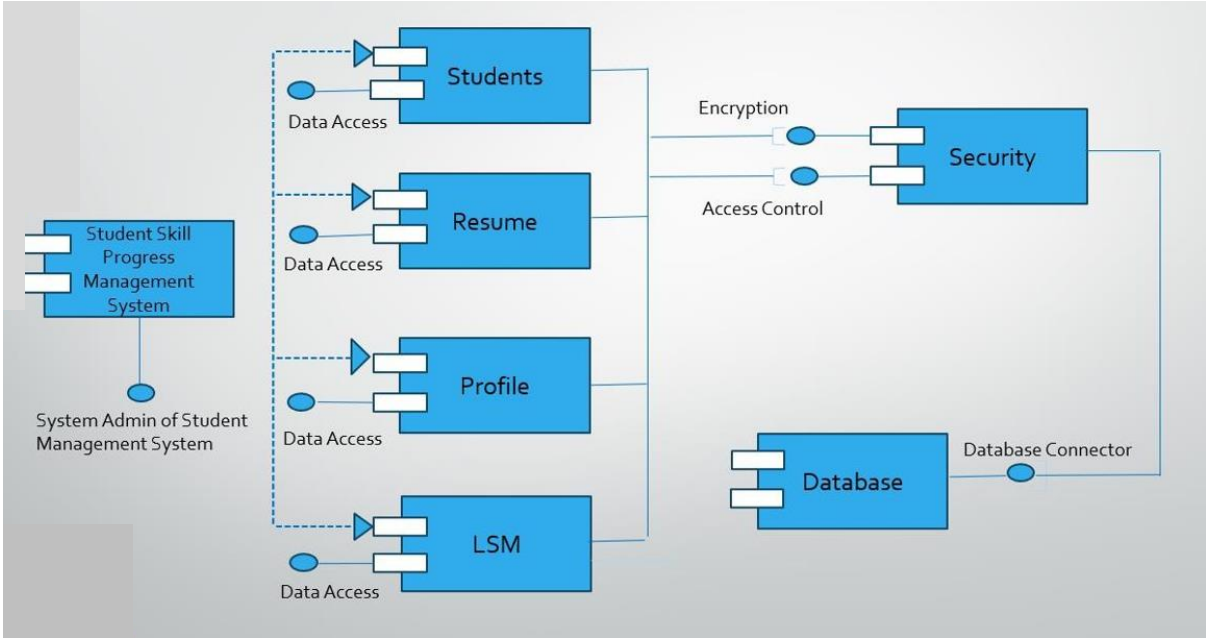

Fig. 3.5 *Component Diagram*

A component diagram in the context of the "Student Skill Progress and Management System" (SSPMS) provides a high-level view of the system's components and their interactions. It focuses on the structural aspects of the system, illustrating the main software components/modules and their relationships.

Student Skill Progress and Management System (SSPMS):

This is the main component representing the entire system.

It encapsulates all the other components and modules, serving as the overarching container for the SSPMS functionality.

Security Component:

Ensures secure access to the SSPMS.

Handles user authentication (login) and authorization (permissions and roles).

Interfaces with user profiles and ensures that only authorized users can access specific features.

Interactive Resume Builder Component:

Facilitates the creation and customization of interactive resumes by students.

Allows students to input their skills, experiences, and achievements, and then generates visually appealing resumes.

Interfaces with the database to store student resume data.

Database Component:

Represents the database system used in SSPMS.

Stores various data, including student profiles, course information, progress records, resume data, and more.

Provides a structured and efficient way to manage and retrieve data.

LMS Component:

This component allows students to manage exam schedules and get reminders of the registered exams.

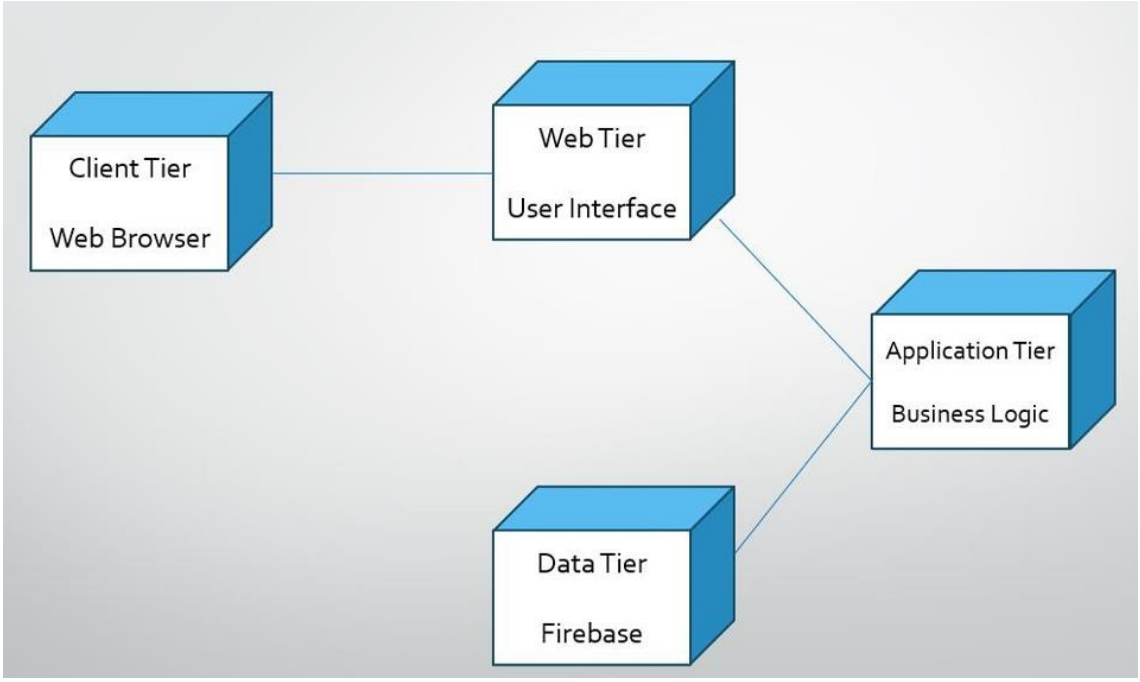

Fig. 3.6 *Deployment Diagram*

The deployment diagram you're describing represents the different tiers or layers in a typical web-based application architecture, illustrating the distribution of components and where they run.

Client Tier (Web Browser):

The client tier represents the user interface, typically rendered in a web browser (e.g., Chrome, Firefox).

Users interact with the application through the web browser, accessing web pages, submitting forms, and interacting with the user interface elements.

This tier is responsible for presenting the application's visual components to users and receiving user input.

Web Tier (User Interface):

The web tier includes the user interface components of the application.

It encompasses the front-end development, which consists of HTML, CSS, JavaScript, and any front-end frameworks (e.g., React.js) used to create the interactive user experience.

The web tier communicates with the application tier (business logic) to request and display data, handle user actions, and update the interface in response to user interactions.

Application Tier (Business Logic):

The application tier contains the core business logic and processing logic of the application.

It handles user requests, processes data, executes application-specific functions, and implements the rules that govern the application's behavior.

Data Tier (Firebase):

The data tier represents the storage and management of data used by the application.

In this specific deployment, Firebase serves as the database and backend service.

Firebase provides cloud-based data storage (Firestore/Realtime Database), authentication services, and other backend functionalities.

The data tier is responsible for storing and retrieving data needed for the application's functionality, including user profiles, course information, progress tracking, interactive resumes, and more.

- The client tier (web browser) interacts with the web tier (user interface) to display the application to users and capture their interactions.
- The web tier communicates with the application tier (business logic) to request data and perform actions.
- The application tier interacts with the data tier (Firebase) to retrieve and store data.

4. IMPLEMENTATION

Module 1: Registration, Login and Student Profile

This module focuses on user authentication, account creation, and managing user profiles within the educational platform. It's a foundational component that ensures security, personalized experiences, and proper identification of users.

Registration*:* New students can create accounts by providing essential information such as name, email, password, and possibly additional details. This information is used to create a user profile.

Login: Users (students, instructors, administrators) can log into the system using their credentials (username/email and password). Authentication mechanisms should be secure to prevent unauthorized access.

Student Profile: Each student has a profile containing personal information, academic history, progress in courses, and other relevant details. Students can update and manage their profiles, providing an opportunity to showcase their achievements and customize the experience.

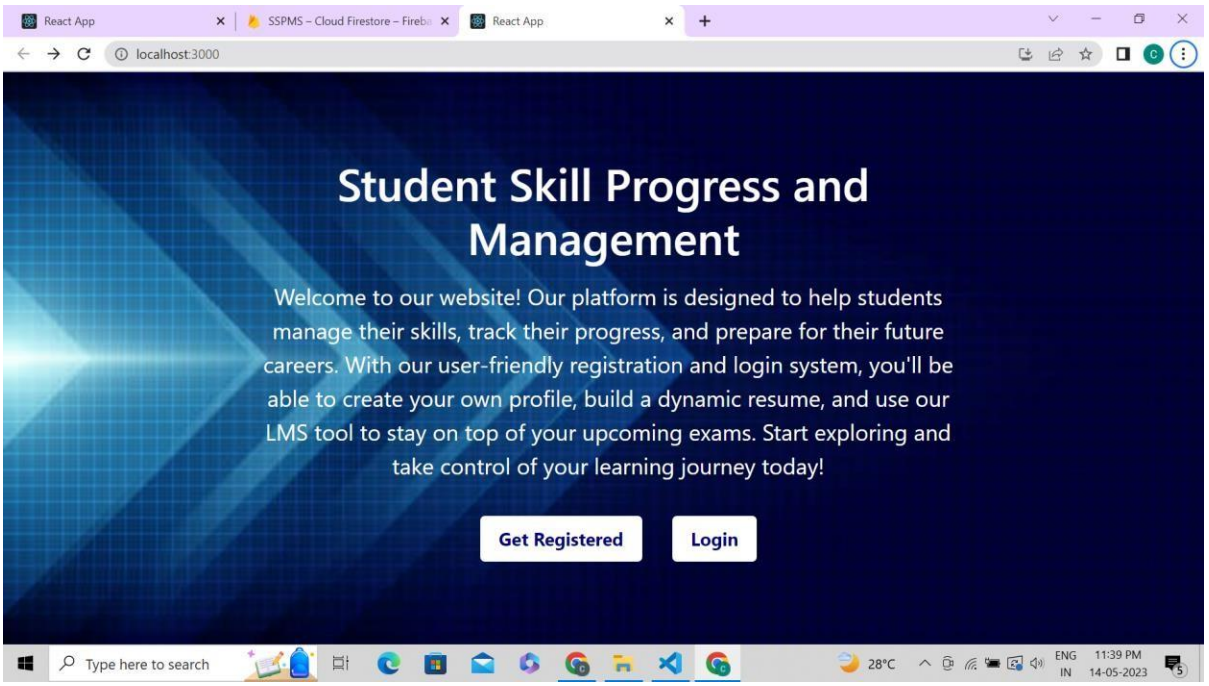

Fig.4.1 *Introduction page* : *Guides the user about the application*

```
const Home = () => {
  return (
    <div>
          ..............
```

```
            ……….
          <Link to="/Register">
            <button className='register-button'>Get Registered</button>
          </Link>
          <Link to='/login'>
            <button className='login-button'>Login</button>
          </Link>
        </div>
      </div>

    </div>
  )
}

export default Home;
```

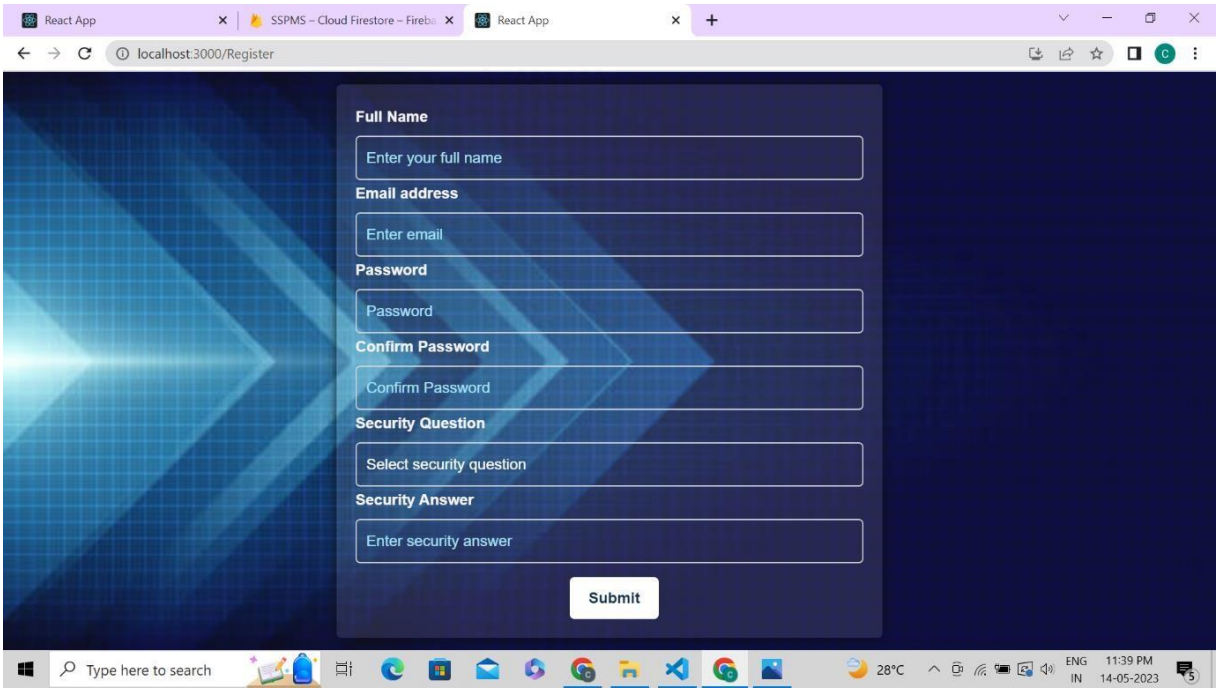

Fig.4.2.1 *Registration page* : *allows users to create new accounts by providing necessary information and credentials.*

```
function Register() {

  const [fullName, setFullName] = useState("");
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [confirmPassword, setConfirmPassword] = useState("");
  const [securityQuestion, setSecurityQuestion] = useState("");
  const [securityAnswer, setSecurityAnswer] = useState("");
```

```
    const navigate = useNavigate()

    const handleSubmit = async (event) => {
      event.preventDefault();
      const salt = await bcrypt.genSalt(10);
      const hashedPassword = await bcrypt.hash(password, salt);

      try {
        await createUserWithEmailAndPassword(auth, email, password);
        await addDoc(collection(database, "students"), {
          fullName,
          email,
          password: hashedPassword,
          securityQuestion,
          securityAnswer,
        });
        console.log("student added")
        navigate('/login')

      } catch (e) {
        console.error("Error adding document: ", e);
        alert("Email already registered");
      }

}
```

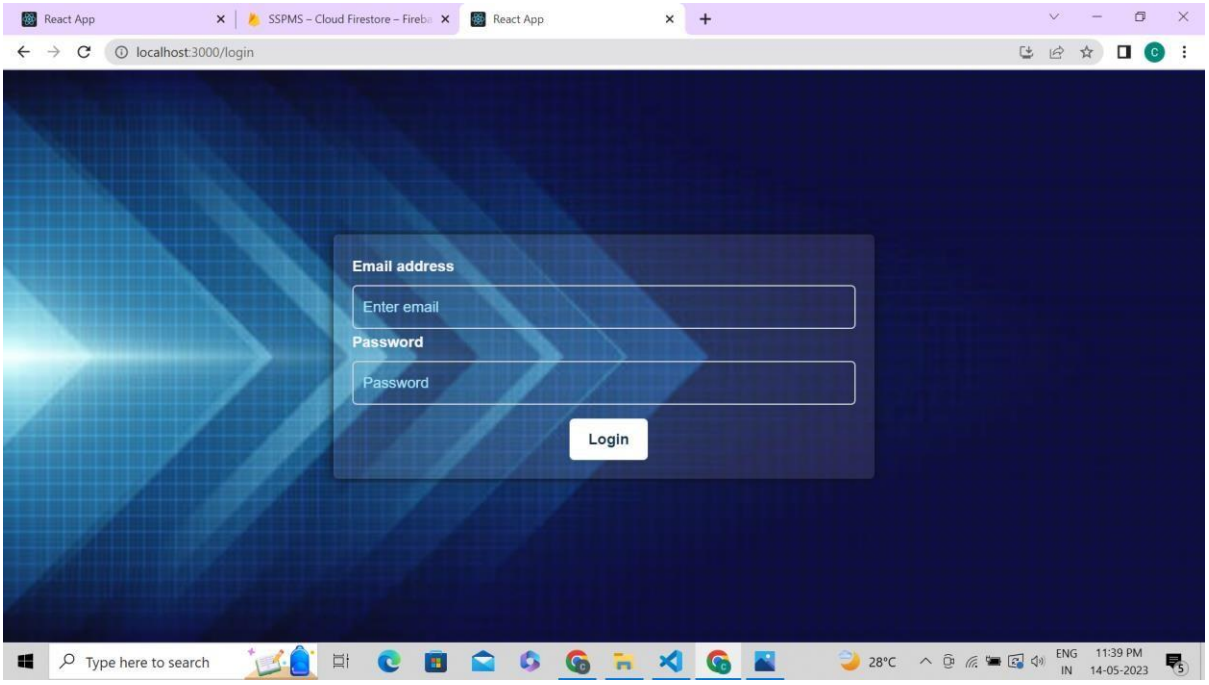

Fig.4.2.2 *Login page* : *verifies user identity and grants access to a system or application based on provided credentials.*

```
const Login  = () => {
const [email, setEmail] = useState("");
const [password, setPassword] = useState("");const
navigate = useNavigate();
const auth = getAuth();


const handleLogin = (event) => { event.preventDefault();
   signInWithEmailAndPassword(auth, email, password)
     .then((userCredential) => { console.log(email)
       localStorage.setItem("userEmail", email);
       navigate("/user",{ state: { userEmail : email } })
     })
     .catch((error) => { console.log(error)
       alert("Incorrect email/password")
     });
};
…….
```

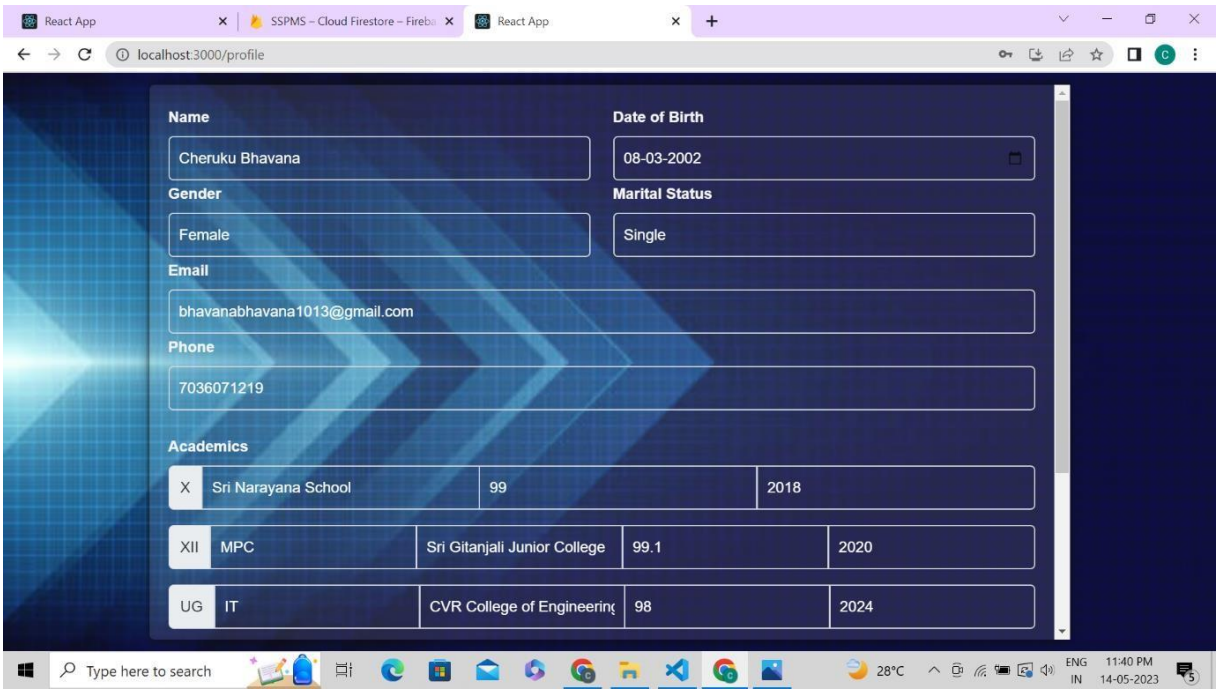

Fig.4.3 *Profile Page* : *Collects the personal and professional information of the user.*

```
const Profile = () => {
   const [formData, setFormData] = useState({
      name: "",
      dob: "",
      gender: "",
      maritalStatus: "",
      email: "",
      phone: "",
      careerObjective: "",
      XInstitute: "",
      XPercentage: "",
      XCompletion: "",
      XIIInstitute: "",
      XIIBranch: "",
      XIIPercentage: "",
      XIICompletion: "",
      UGInstitute: "",
      UGBranch: "",
      UGPercentage: "",
      UGCompletion: "",
      skills: "",
   });
const navigate = useNavigate()
const userEmail = localStorage.getItem("userEmail");
.....
export default  Profile
```

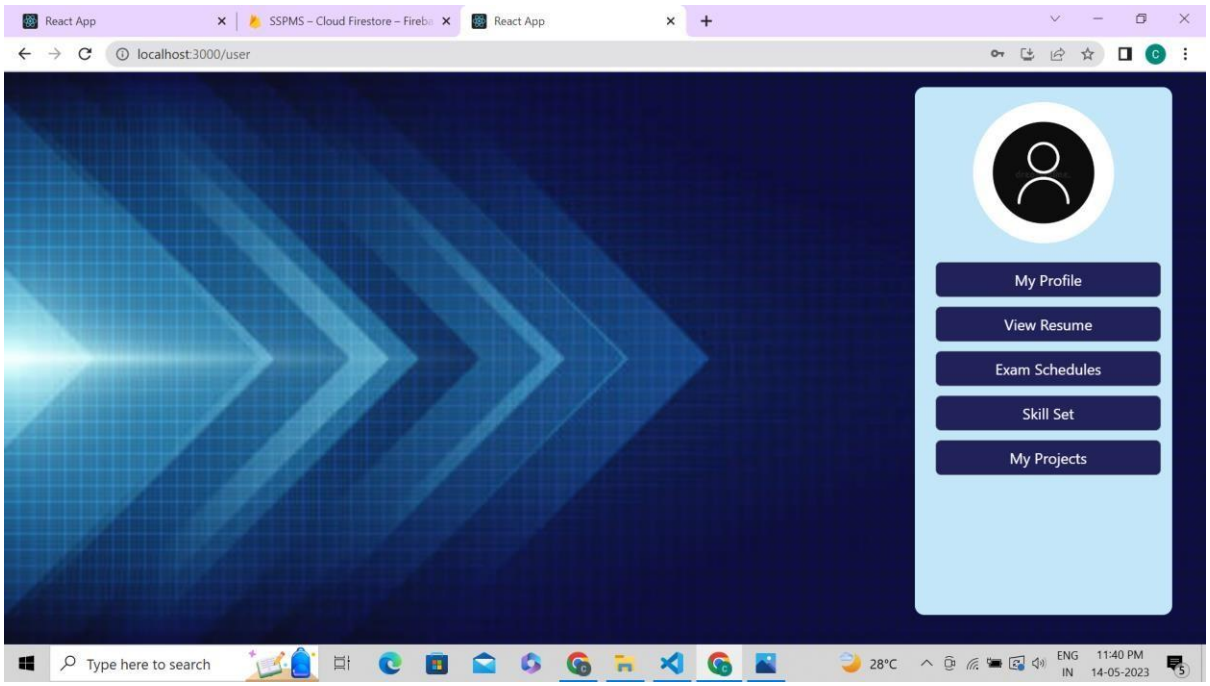

Fig.4.4 *Dashboard*: *an overview of key information and data, allowing users to monitor and analyzeimportant metrics at a glance*

```
const User = () => {
const navigate = useNavigate();const
location = useLocation();
const userEmail = location.state.userEmail

const handleProfileClick = () => {
  navigate('/profile');
};

const handleAddExamClick = () => {
  navigate('/exams');
};
const handleResumeClick = () => {
  navigate('/resume')
}
const handleProjectsClick = () => {
  navigate('/projects')
}

const handleSoftSkillsClick = () => {
  navigate('/skills')
} ……
```

Module 2: Interactive Resume Builder

The Interactive Resume Builder module empowers students to create professional and dynamic resumes. It goes beyond traditional static resumes by providing customization and interactivity.

Resume Creation: Students can create resumes by entering details about their education, work experience, skills, projects, certifications, and other relevant information.

Customization: Allow students to customize their resumes by adding, removing, or reordering sections.

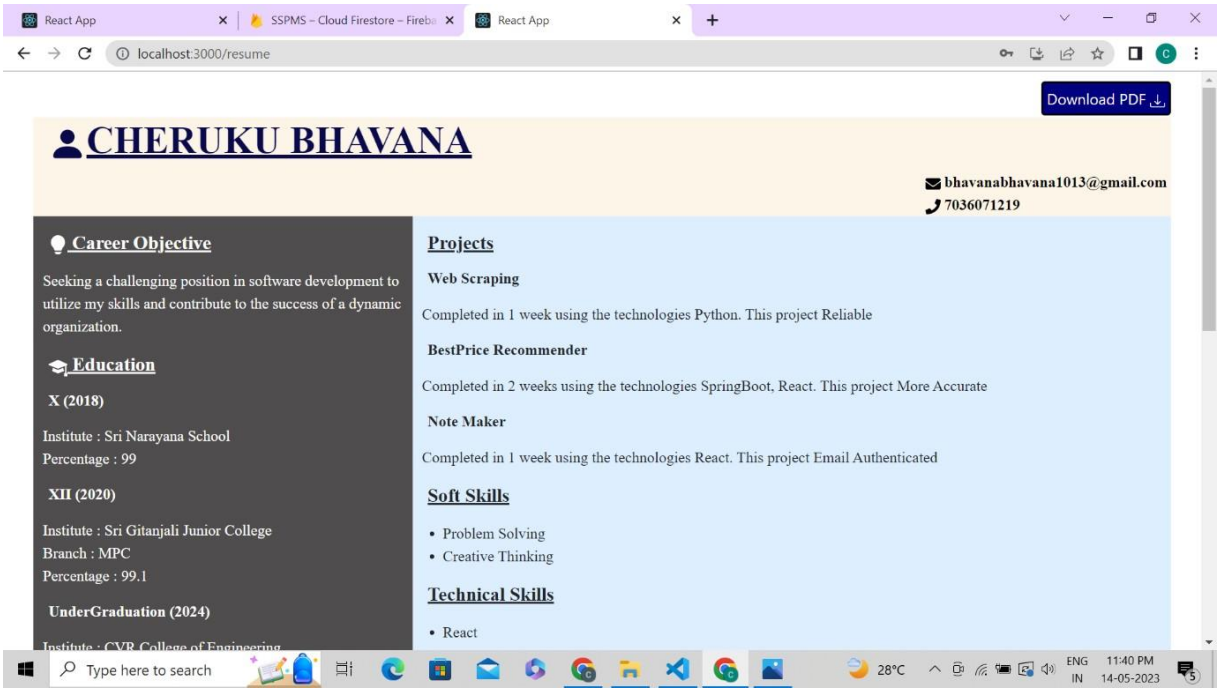

Fig.4.5 *Dynamic resume generator: uses user input to automatically create a personalized resume, adapting its content based on the provided information*

```
import React from 'react' import
'../css/resume.css'
import { useState, useEffect, useRef } from 'react'; import {
collection, getDocs } from 'firebase/firestore';import {
database } from "../firebase";
import html2pdf from 'html2pdf.js';
import 'material-icons/iconfont/material-icons.css' import {
FaPhone, FaEnvelope, } from 'react-icons/fa';
import { BsFillFileEarmarkPdfFill, BsLinkedin, BsTrophyFill, BsFillLightbulbFill,
BsFillPersonFill, BsFillMortarboardFill, BsAward, BsArrowDownCircleFill,
BsDownload } from "react-icons/bs";
```

```
const Resume = () => {
  const userEmail = localStorage.getItem("userEmail");console.log(userEmail)
  const [details, setDetails] = useState({name: '',
    email: '',
    phone: '', careerObjective: '',
    XInstitute: '', XPercentage: '',
    XIIInstitute: '', XIIPercentage:
    '', XIIBranch: '', UGInstitute: '',
    UGPercentage: '',UGBranch: '',
    projects: [],
    XCompletion: '',
    XIICompletion: '',
    UGCompletion: '',
    skills:[],
    Tskills:[]
  })
```

Module 3: Learning Management System (LMS) Tools

The LMS Tools module facilitates the administration, delivery, and tracking of educational content and student progress.

Here, Students get reminders of the examinations that are scheduled.

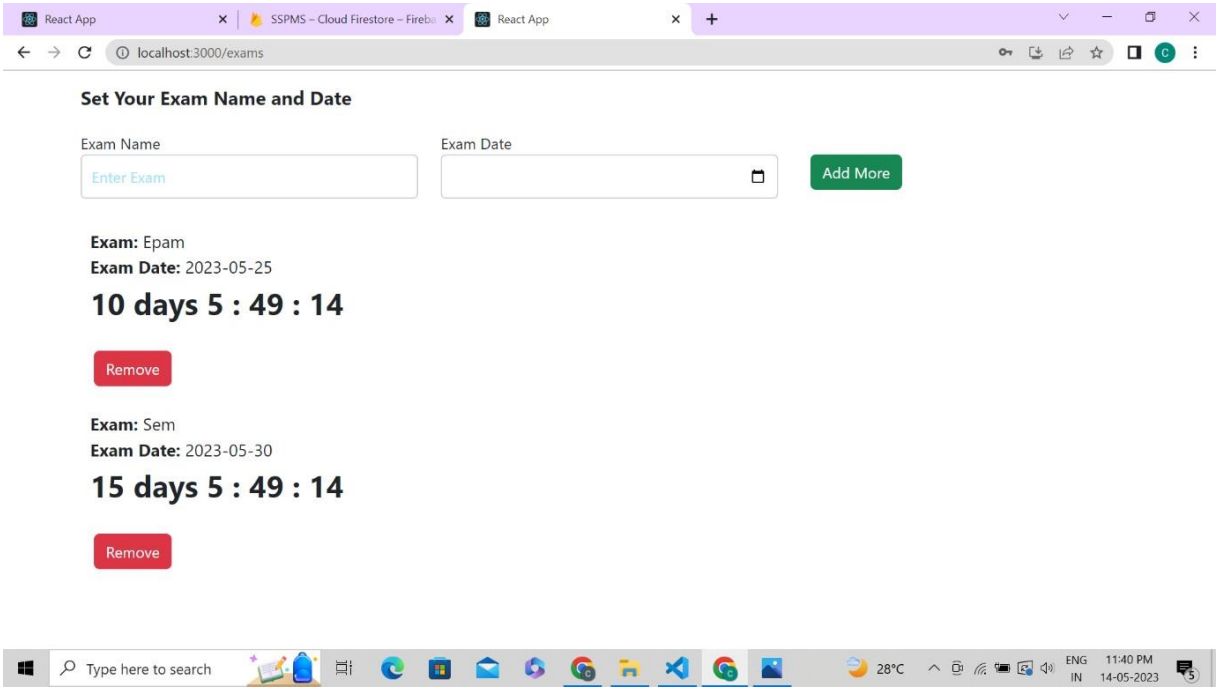

Fig.4.6 *LMS countdown for exams enables the scheduling of exams by displaying a countdown timer, ensuring timely start and completion of preparation for learners.*

```
function Exam() {
 const [inputList, setinputList] = useState([{ examName: '', examDate: '' }]);const
 [exams, setExams] = useState([])
 const [remainingTime, setRemainingTime] = useState({ days: 0, hours: 0, minutes:0,
seconds: 0 });
 const userEmail = localStorage.getItem('userEmail')

 useEffect(() => {
  const fetchExams = async () => {try {
    if (userEmail) {
     const collectionRef = collection(database, "basic_details");const
     querySnapshot = await getDocs(collectionRef);

     for (const doc of querySnapshot.docs) {if
      (doc.data().email === userEmail) {
       const examsData = doc.data().exams || [];;
       setExams(examsData);
       break;
      }
     }
    }
   }
   catch (error) { console.log(error);
   }


  };

  fetchExams();
 }, [userEmail]);
```

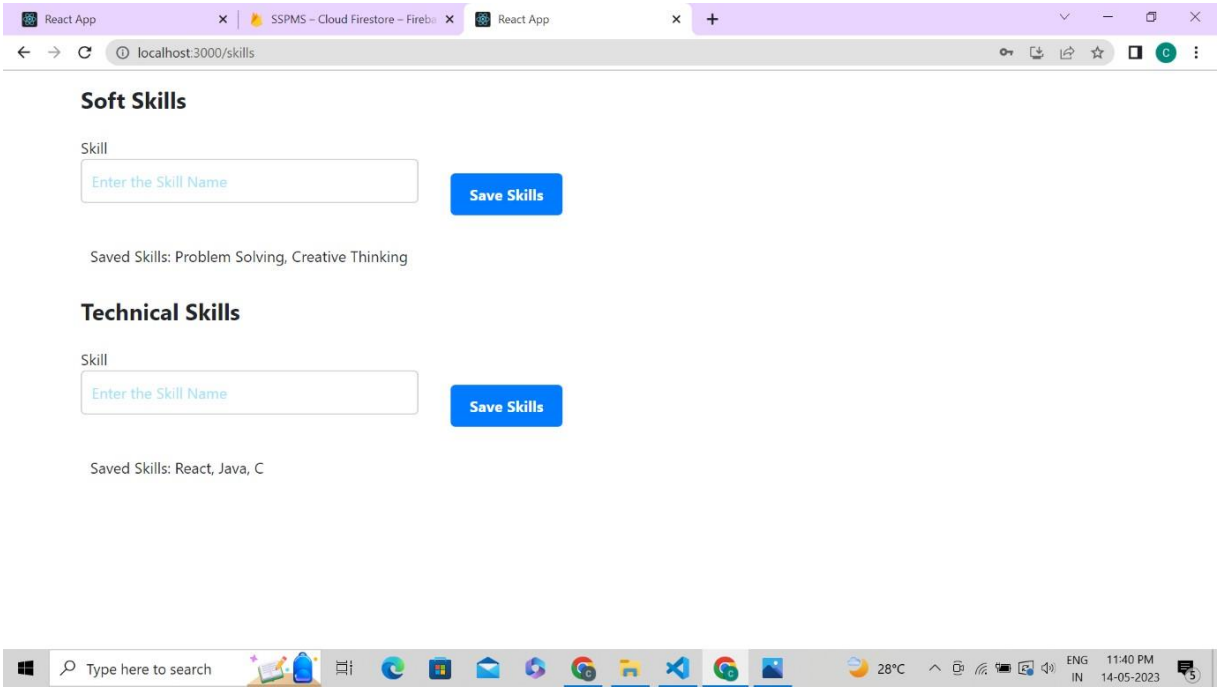

Fig.4.7 *Collecting both user soft skills (such as communication, teamwork, leadership) and technical skills (such as programming languages, software proficiency) helps create a comprehensive profile for effective job matching or skill development.*

```
function Skills() {
  const [inputList, setinputList] = useState([{ Skill: '' }]);const [skills,
  setSkills] = useState([])
  const [TinputList, setTinputList] = useState([{ TSkill: '', noOfModules: '',
noOfModulesCompleted: '' }]);
  const [Tskills, setTSkills] = useState([])
  const userEmail = localStorage.getItem('userEmail')

  useEffect(() => {
        const fetchSkills = async () => {
          try {
            if (userEmail) {
              const collectionRef = collection(database, "basic_details");
              const querySnapshot = await getDocs(collectionRef)
```

```
      for (const doc of querySnapshot.docs) {
        if (doc.data().email === userEmail) {
        const skillsData = doc.data().skills;
        setSkills(skillsData);
          const TskillsData = doc.data().Tskills;
          setTSkills(TskillsData);
          break;
        }
      }
    }
  }
   catch (error) {
    console.log(error);
  }
 };
 fetchSkills();
}, [userEmail]);
```

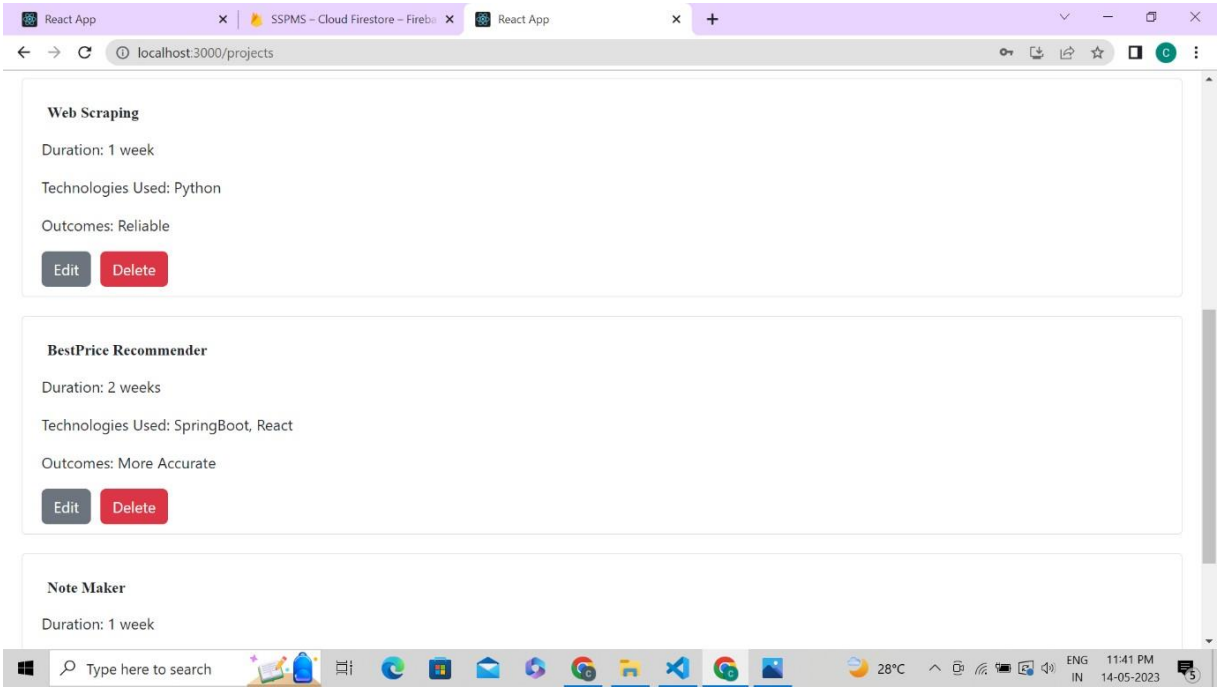

Fig.4.8 *Allowing users to modify, delete, and add new skills ensures a dynamic and up-to-date skillprofile, reflecting their evolving abilities and enhancing accuracy in skill representation.*

```
 const LMS = () => {
const endTime = new Date('June 30, 2023 00:00:00').getTime();
const [currentTime,setcurrentTime] = useState(new Date().getTime());const
gap = endTime - currentTime;

const seconds = 1000;
const minutes = seconds * 60;const
hours = minutes * 60; const days =
hours * 24;

const remainingDays = Math.floor(gap / days);
const remainingHours = Math.floor( (gap % days) / hours); const
remainingMinutes = Math.floor( (gap % hours) / minutes);
const remainingSeconds = Math.floor( (gap % minutes) / seconds);

useEffect(()=>{
  setTimeout(()=>setcurrentTime(new   Date().getTime()),1000);
},[currentTime])

return (
  <div>
    <center>

      <Display days={remainingDays} hours={remainingHours}
      minutes={remainingMinutes}seconds={remainingSeconds}
      />

    </center>
  </div>
 )
}
export default LMS;
```

5. TESTING

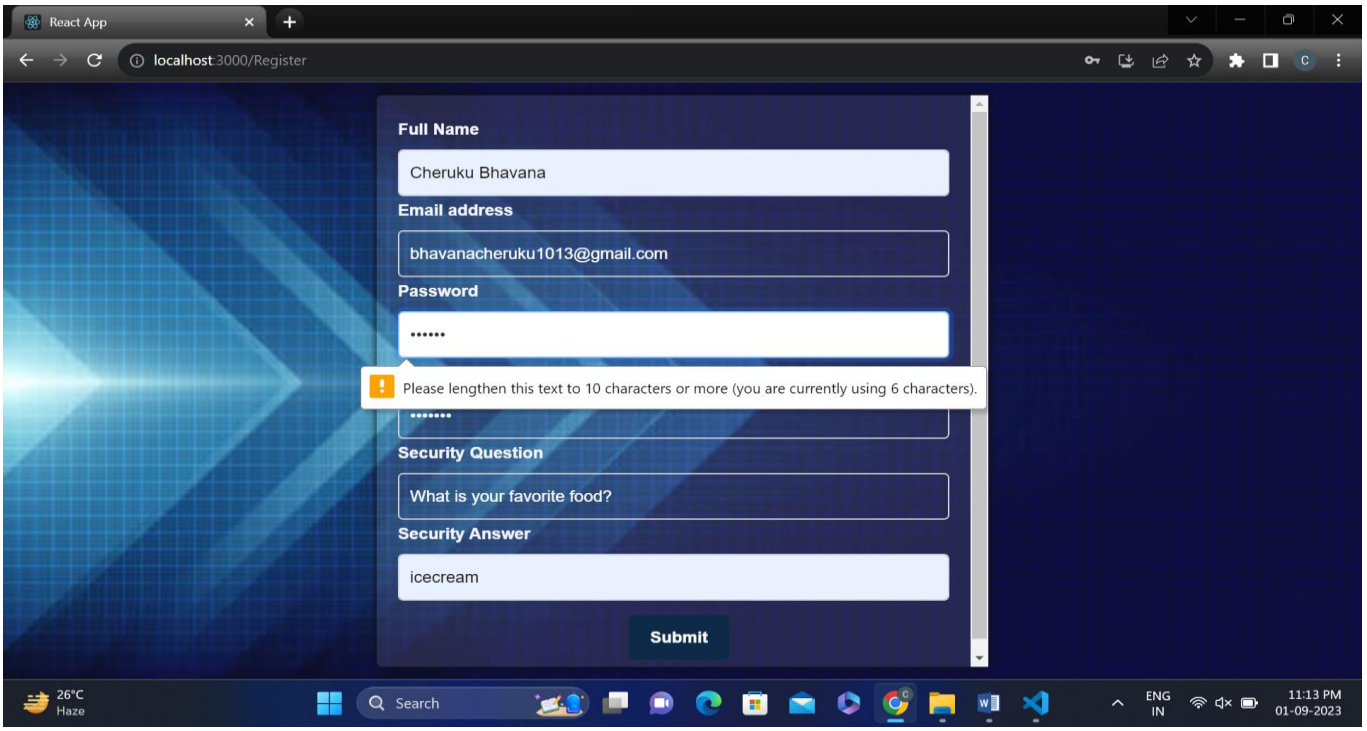

Fig.5.1 *Registration page alerting the user to enter valid password*

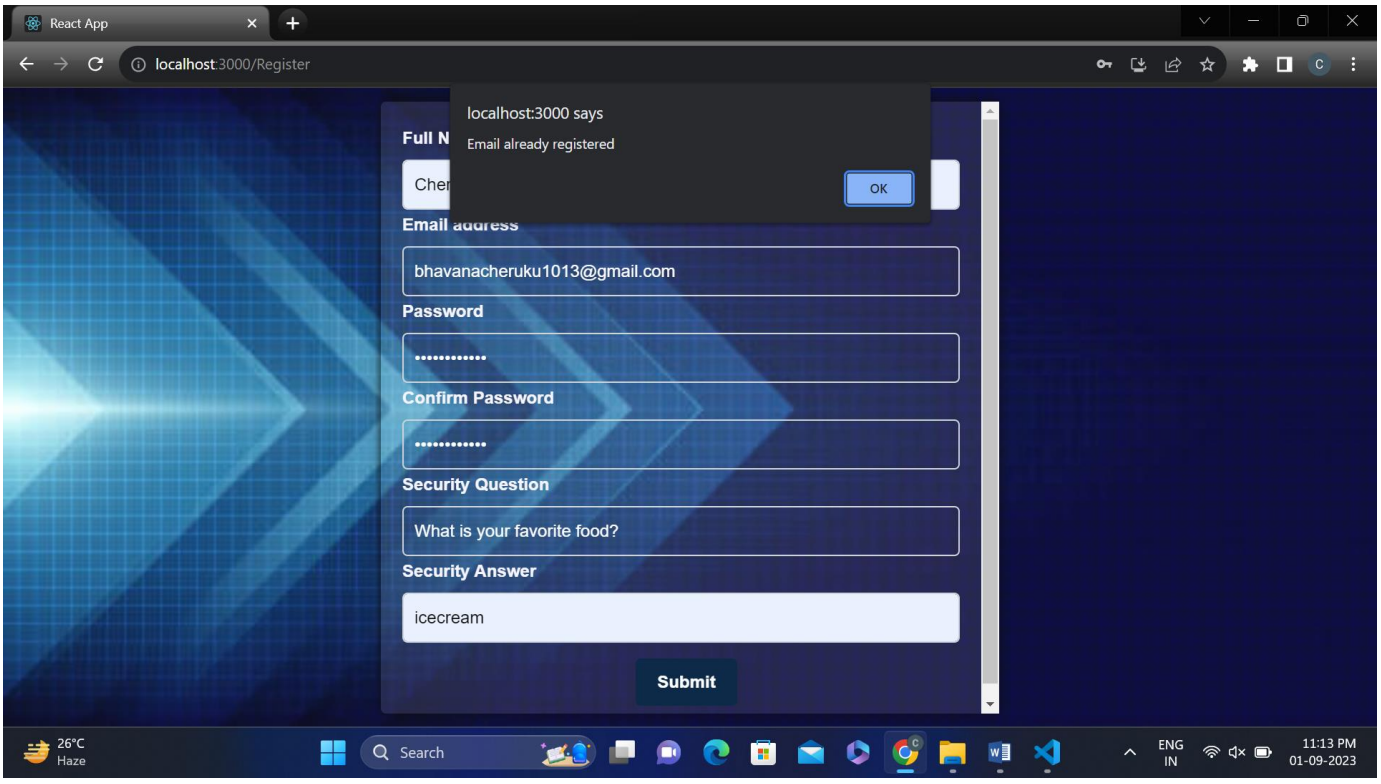

Fig.5.2 *Registration page alerting that the user had already registered with that email*

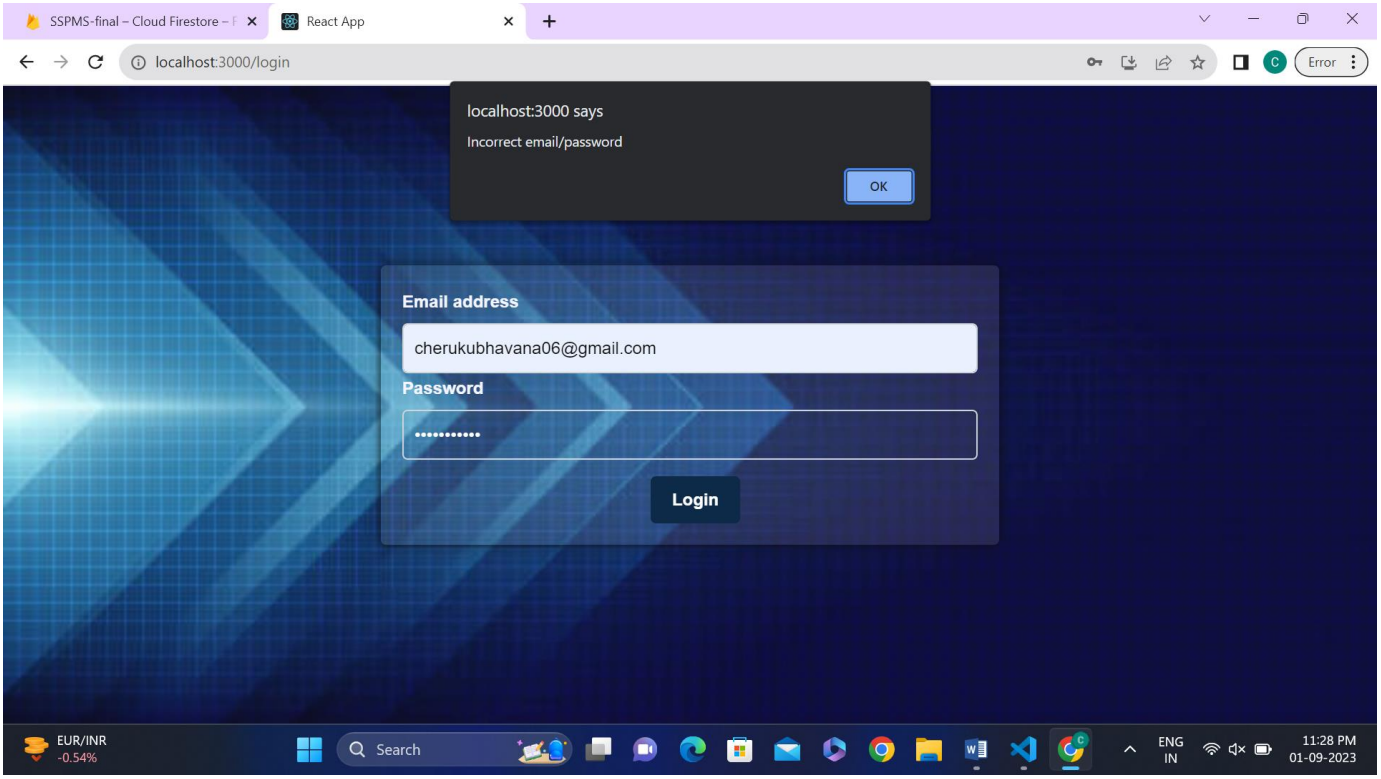

Fig.5.3 *Registration page alerting the user to enter valid email/password.*

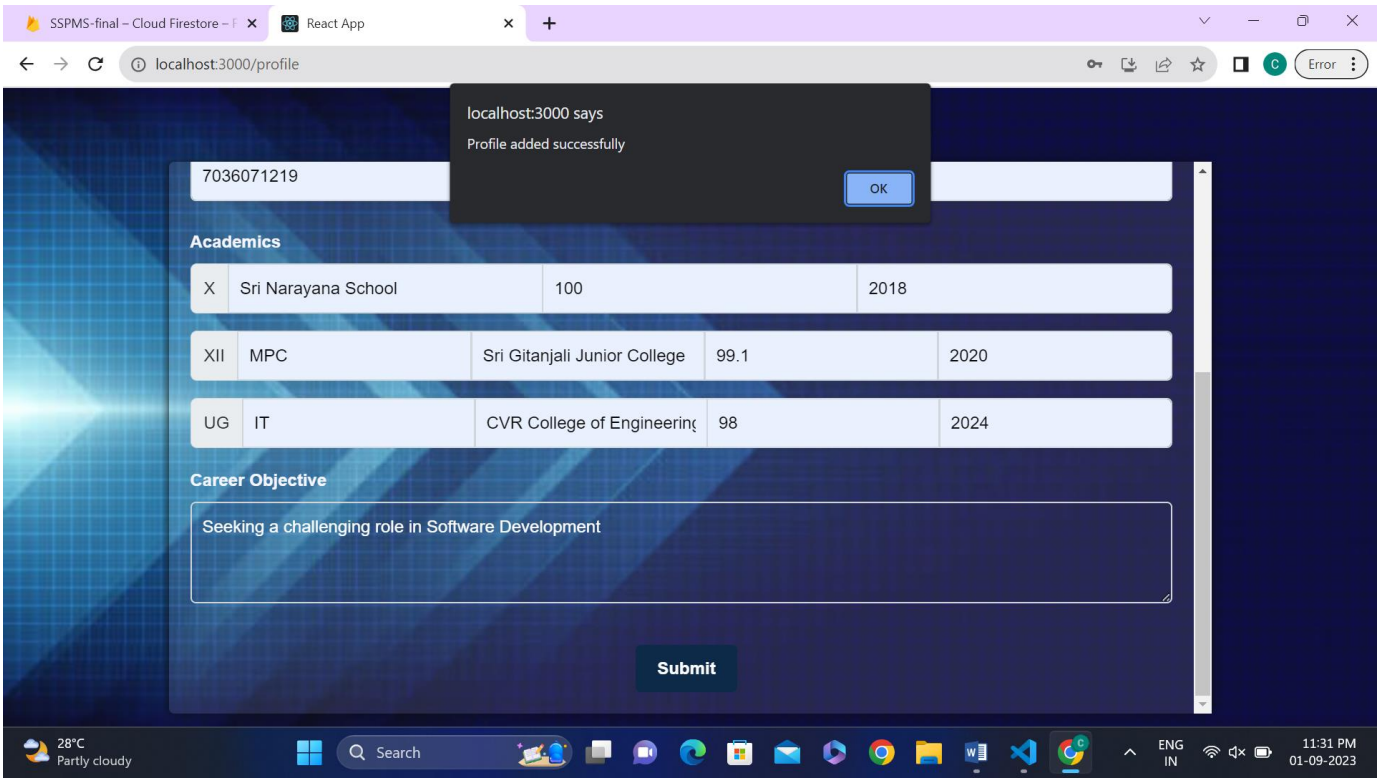

Fig.5.4 *User Profile is successfully added*

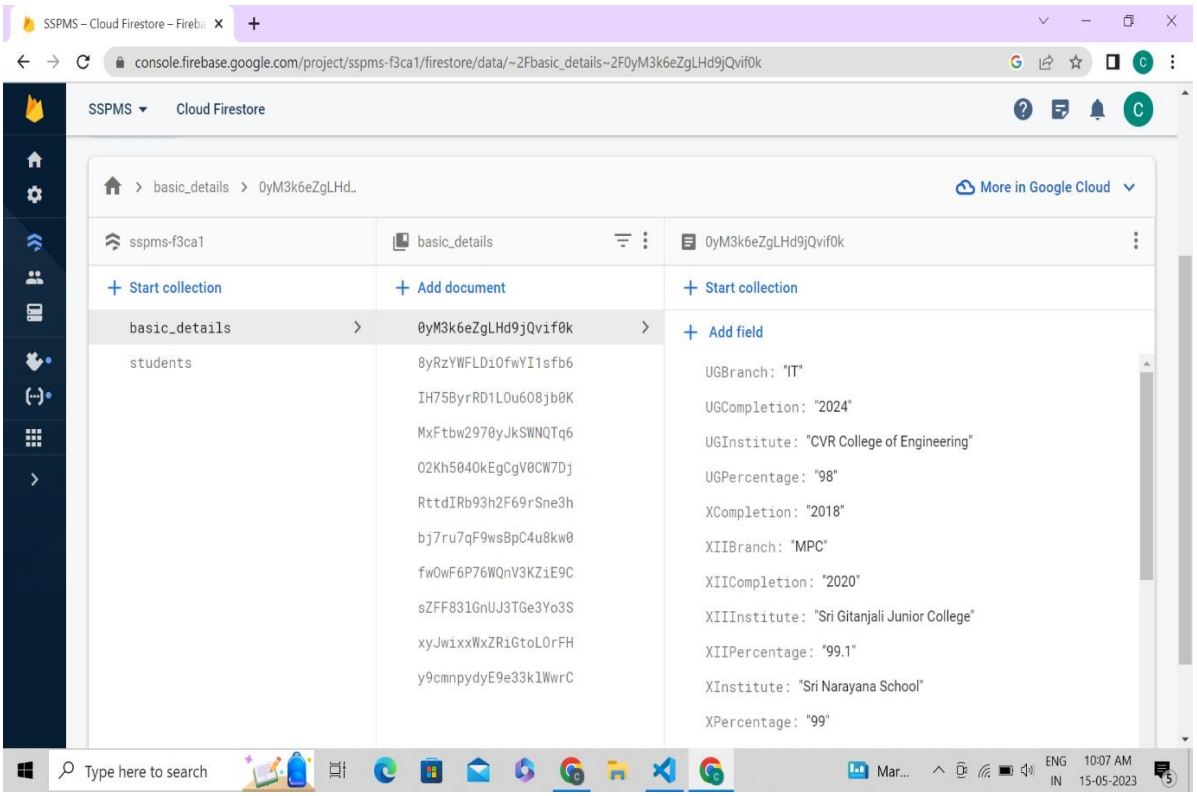

Fig.5.5 *Storing user details in Firebase, a cloud-based platform, offers secure and scalable data storage, real-time synchronization, and easy accessibility for applications requiring user-related information.*

CONCLUSION

In conclusion, the Student Management Project aims to provide a holistic and user-centric platform that empowers students with valuable tools for career development and facilitates efficient learning management. Through the Login, Registration, and Student Profile module, users gain secure access to the platform, can personalize their profiles, and engage with a personalized experience. The Interactive Resume Builder module enhances students' professional branding by enabling them to create dynamic and customizable resumes, setting the stage for their career success. The Learning Management System (LMS) Tools module ensures effective course management, progress tracking, and performance analytics, creating a supportive environment for both students and instructors.

This project recognizes the importance of user-friendliness, security, and compatibility, aligning with modern industry standards and best practices. By adhering to these principles and providing dynamic features, the project strives to enrich the educational journey of students, preparing them for the challenges of the professional world.

Through this comprehensive system, we seek to bridge the gap between education and industry, fostering a seamless transition from the classroom to the workforce. The synergy of these modules forms a powerful solution that not only facilitates learning but also equips students with the tools to showcase their skills and accomplishments effectively. By embracing innovation, interactivity, and data-driven insights, this project has the potential to shape the future of student management, bringing efficiency, empowerment, and growth to educational institutions and students alike.

FUTURE ENHANCEMENTS

The Student Management Project has a solid foundation, but there are several areas for future enhancements to further improve the platform's functionality, user experience, and impact.

Career Guidance and Job Matching: Integrate a feature that offers personalized career guidance, job market trends, and job matching based on students' skills and interests. This feature can help students make informed career decisions and connect with relevant job opportunities. It is provided once the student uploads certificates after the completion of their courses.

Availability of different templates for Resume creation: This helps students to choose desired template of desired color and fonts.

Provide resources for the ongoing learning skills: It aims at providing the required material for the students to excel in the learning process.

The Student Skill Progress and Management System (SSPMS) has the potential for significant future scope and growth. Here are some potential areas for future development and expansion:

Integration with Educational Institutions: SSPMS can integrate with schools, colleges, and universities to provide a unified platform for both educators and students. This integration can include features such as class schedules, assignment submissions, and grade tracking.

Analytics and Insights: Enhance the analytics capabilities of SSPMS to provide students and institutions with insights into learning patterns, career trends, and skill gaps. This data can be used for continuous improvement.

User-generated Content: Allow students to contribute resources, study materials, and reviews, creating a collaborative learning environment.

Feedback Mechanism: Create a feedback system for students to suggest improvements and new features, ensuring that SSPMS continues to meet their evolving needs.

Partnerships: Collaborate with educational institutions, industry organizations, and technology providers to expand the reach and capabilities of SSPMS.

REFERENCES

[1] A. Balderas, L. De-La-Fuente-Valentin, M. Ortega-Gomez, J. M. Dodero and D. Burgos, "Learning Management Systems Activity Records for Students' Assessment of Generic Skills," in IEEE Access, vol. 6, pp. 15958-15968, 2018, doi: 10.1109/ACCESS.2018.2816987.

[2] Anagha Vaidya, Rajashree Jain, Prafulla Bafna., Influence of Staff Student Interaction on Student Engagement, IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), 2017. 2. Dastgir Pojee, Farooq Shaikh, Vishal Kuvar, Fahim Rarh, Mohd. Abbas Meghani., Multi-Platform College Management Framework, Proceedings of the 2nd International Conference on Communication and Electronics Systems (ICCES), 2017.

[3]. Fadi Mohsen, Mohamed Shehab Proposing and Testing New Security Cue Designs for OAuth-WebViewEmbedded Mobile Applications, IEEE 3rd International Conference on Collaboration and Internet Computing, 2017.

[4]. Jinny Rhee, Sheri Sheppard, Samantha Brunhaver, Cheryl Carrico, Ruth Streveler Supporting Student Career Development of Undergraduate Engineering, IEEE, 2017.

[5]. Patrick Hung, Jeanne Lam, Chris Wong, Tyrone Chan., A Study on Using Learning Management System with Mobile App, International Symposium on Educational Technology, 2015.

[6]. Tetsuro Kakeshita, Satoshi Yamashita., A Requirement Management Education Support Tool for Requirement Elicitation Process of REBOK, 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence, 2015.

[7]. Wellington Oliveira, Renato Oliveira, Fernando Castor A Study on the Energy Consumption of Android App Development Approaches, IEEE/ACM 14th International Conference on Mining Software Repositories (MSR), 2017.

[8]. React Js Documentation, by Facebook https://reactjs.org/docs/getting-started.html 9. Firebase documentation, by Google https://firebase.google.com/docs.

[9] D. Vukadin, A. S. Kurdija, G. Delač and M. Šilić, "Information Extraction From Free-Form CV Documents in Multiple Languages," in IEEE Access, vol. 9, pp. 84559-84575, 2021, doi: 10.1109/ACCESS.2021.3087913.

[10] M. Maphosa, W. Doorsamy and B. S. Paul, "Student Performance Patterns in Engineering at the University of Johannesburg: An Exploratory Data Analysis," in IEEE Access, vol. 11, pp. 48977-48987, 2023, doi: 10.1109/ACCESS.2023.3277225.

[11] A multilayer prediction approach for the student cognitive skills measurement

Sadique Ahmad, Kan Li, Adnan Amin, Muhammad Shahid Anwar, Wahab Khan

IEEE Access 6, 57470-57484, 2018

APPENDIX – A : ABBREVIATIONS

API: Application Programming Interface

CRUD: Create, Read, Update, Delete (basic database operations)

CSS: Cascading Style Sheets

DB: Database

HTML: Hypertext Markup Language

JS: JavaScript

LMS: Learning Management System

SSPMS: Student Skill Progress and Management System

UI: User Interface

UX: User Experience

APPENDIX – B : SOFTWARE INSTALLATION PROCEDURE

B.1 Operating System – Windows 10 / macOS / Linux

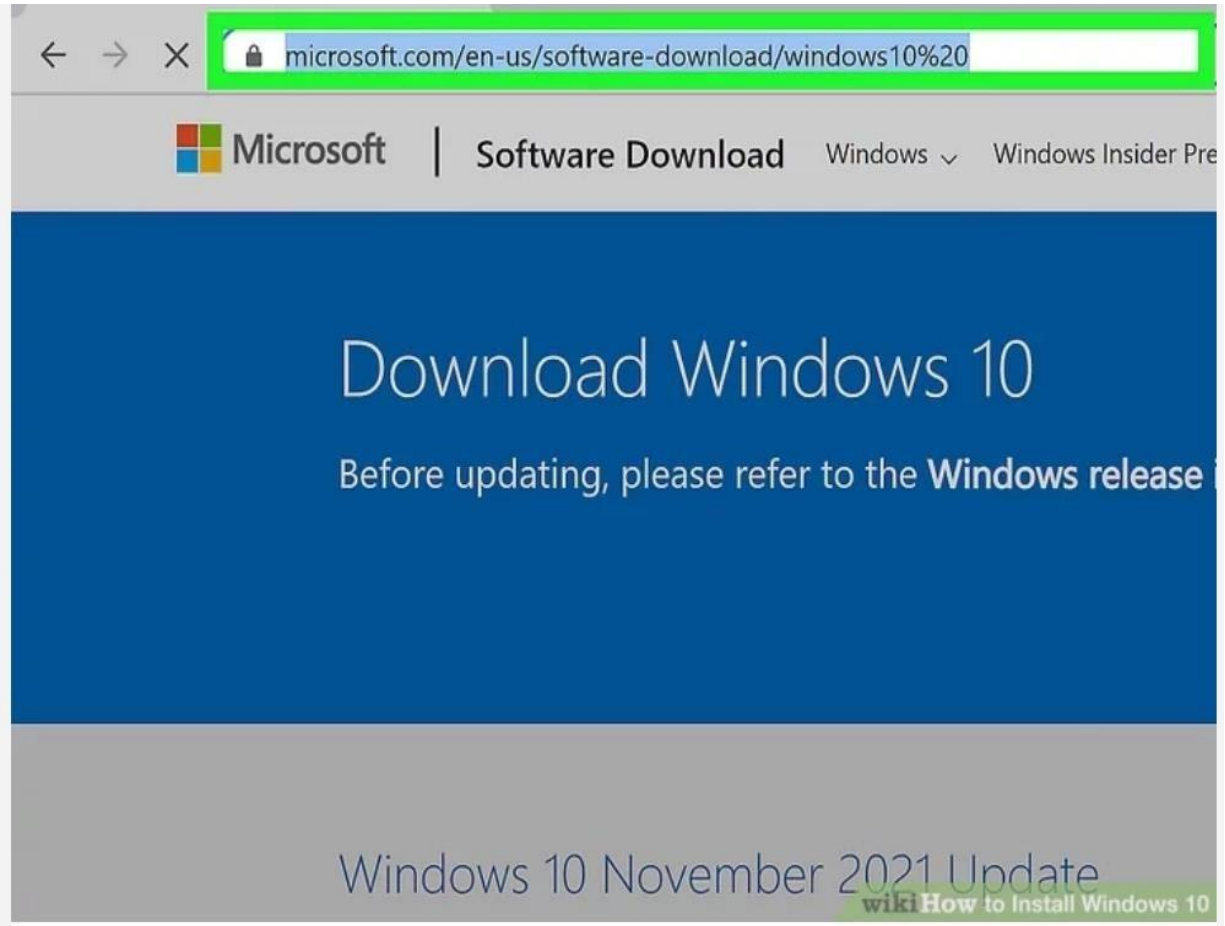

Fig. B.1.1 *Webpage for downloading Windows 10*

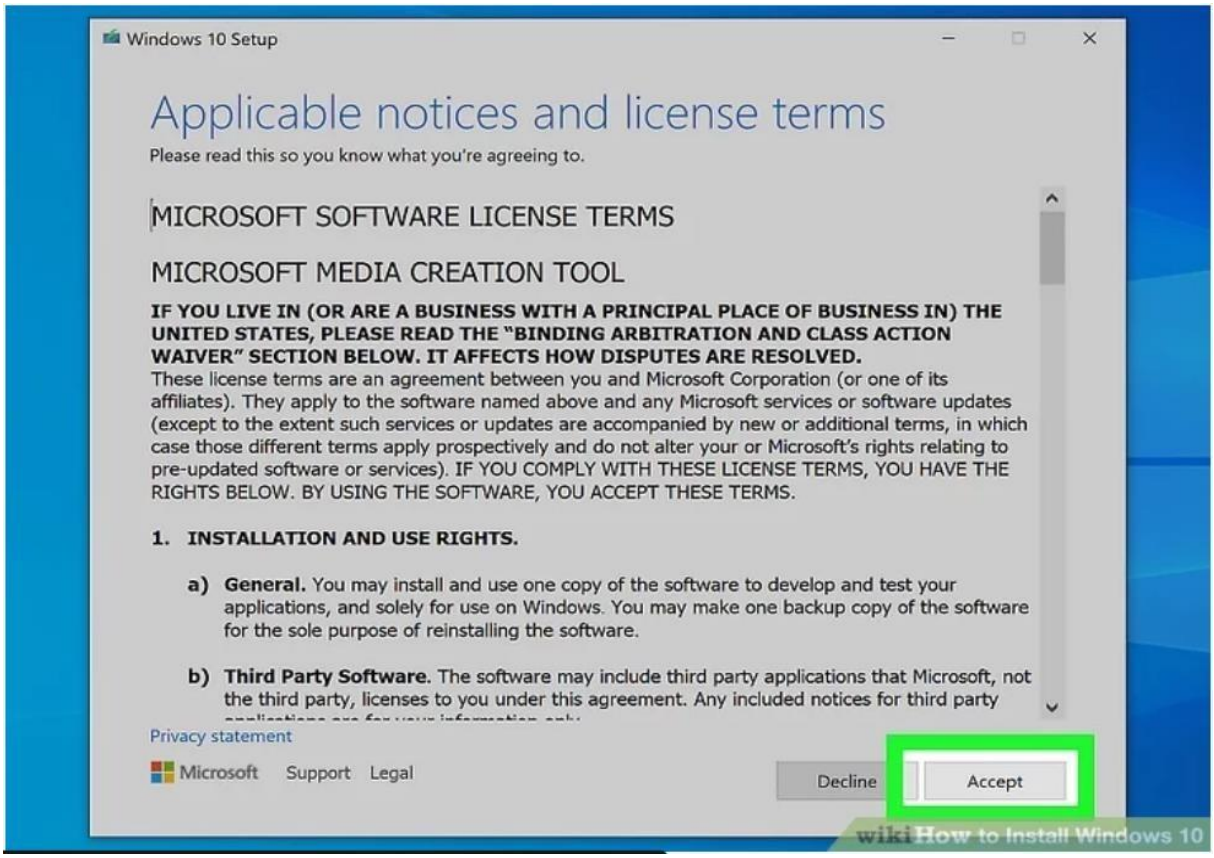

Fig. B.1.2 *Windows 10 Setup*

B.2 Web Browser – Chrome Installation

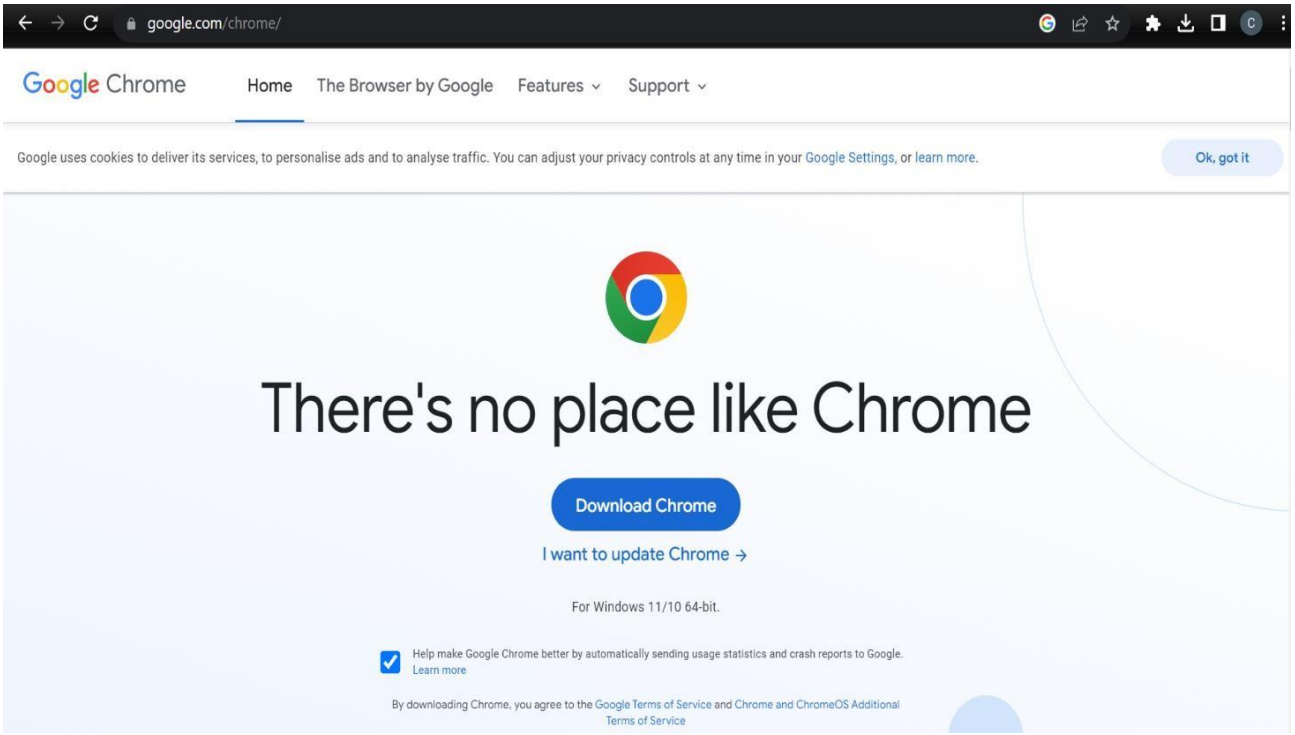

Fig. B.2 *Source: google.com/chrome*

B.3 IDE – Visual Studio Code Installation

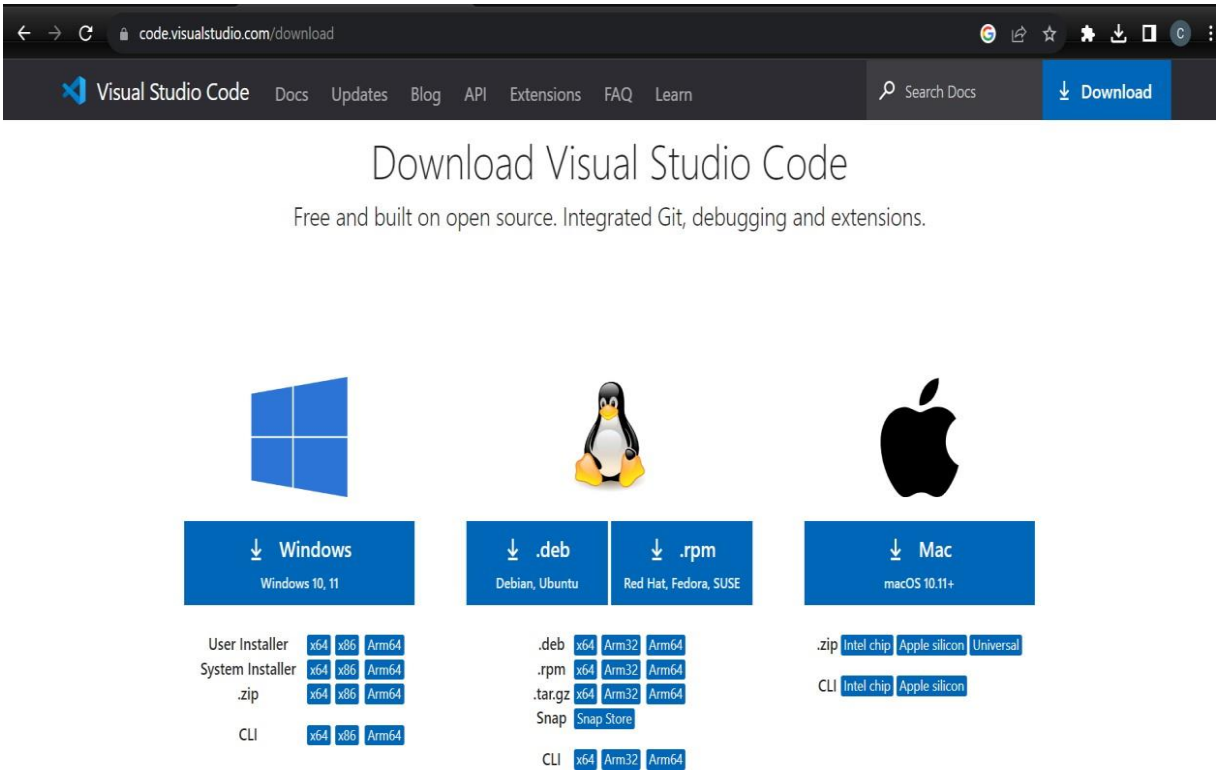

Fig. B.3.1 *Source to download VS Code : code.visualstudio.com/download*

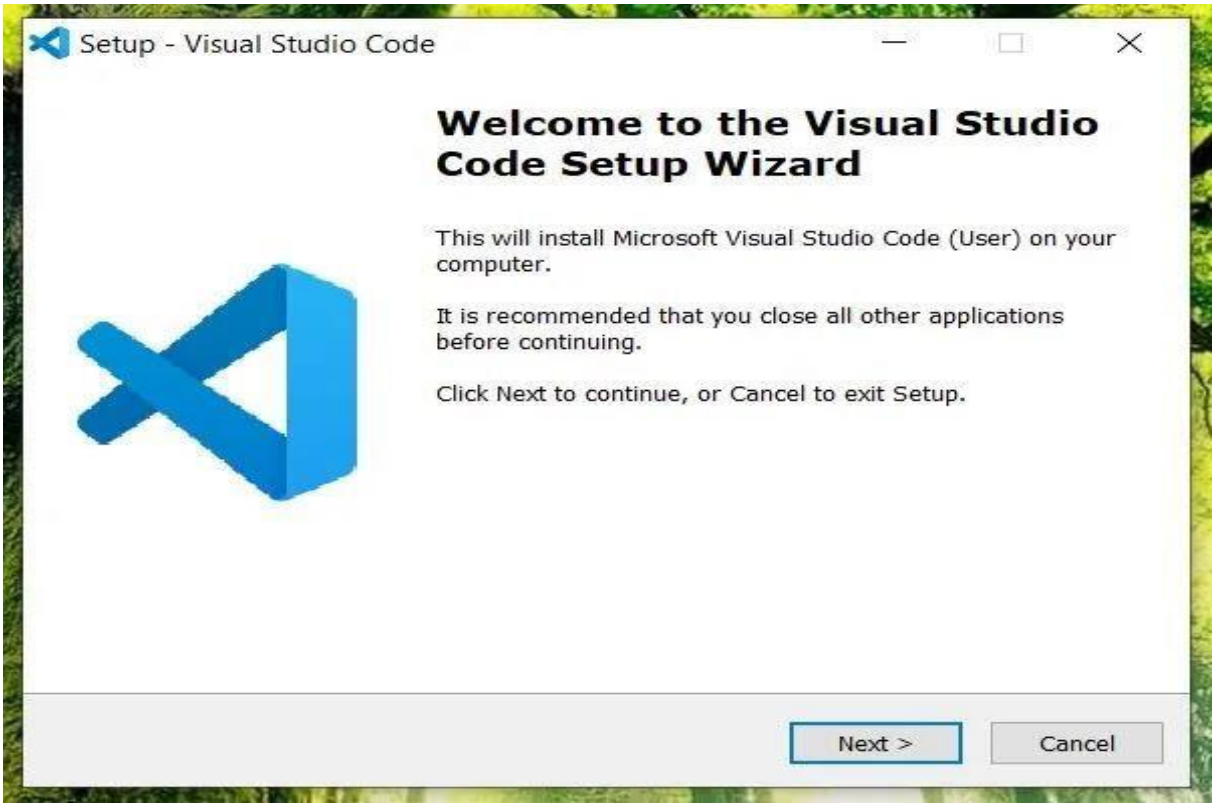

Fig. B.3.2 *VS Code Setup*

B.4 Node.js and NPM

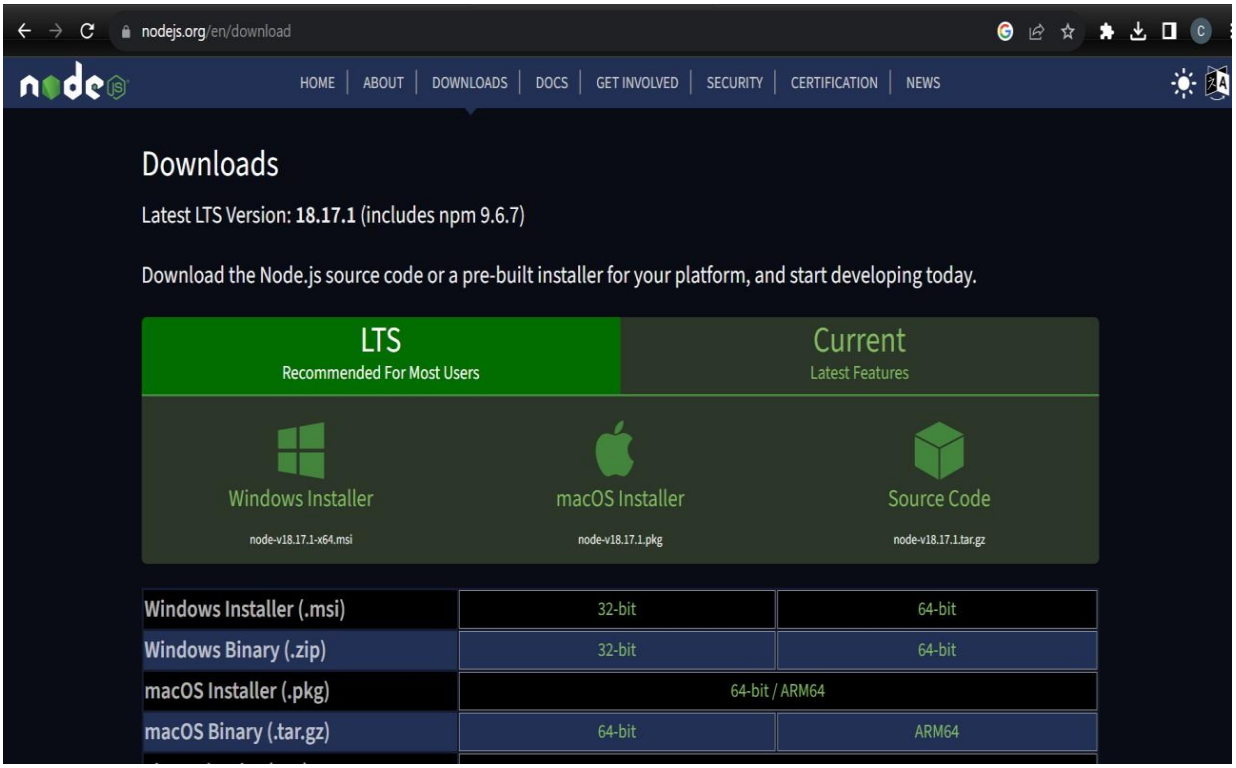

Fig. B.4.1 *Source to download Node.js : nodejs.org/en/download*

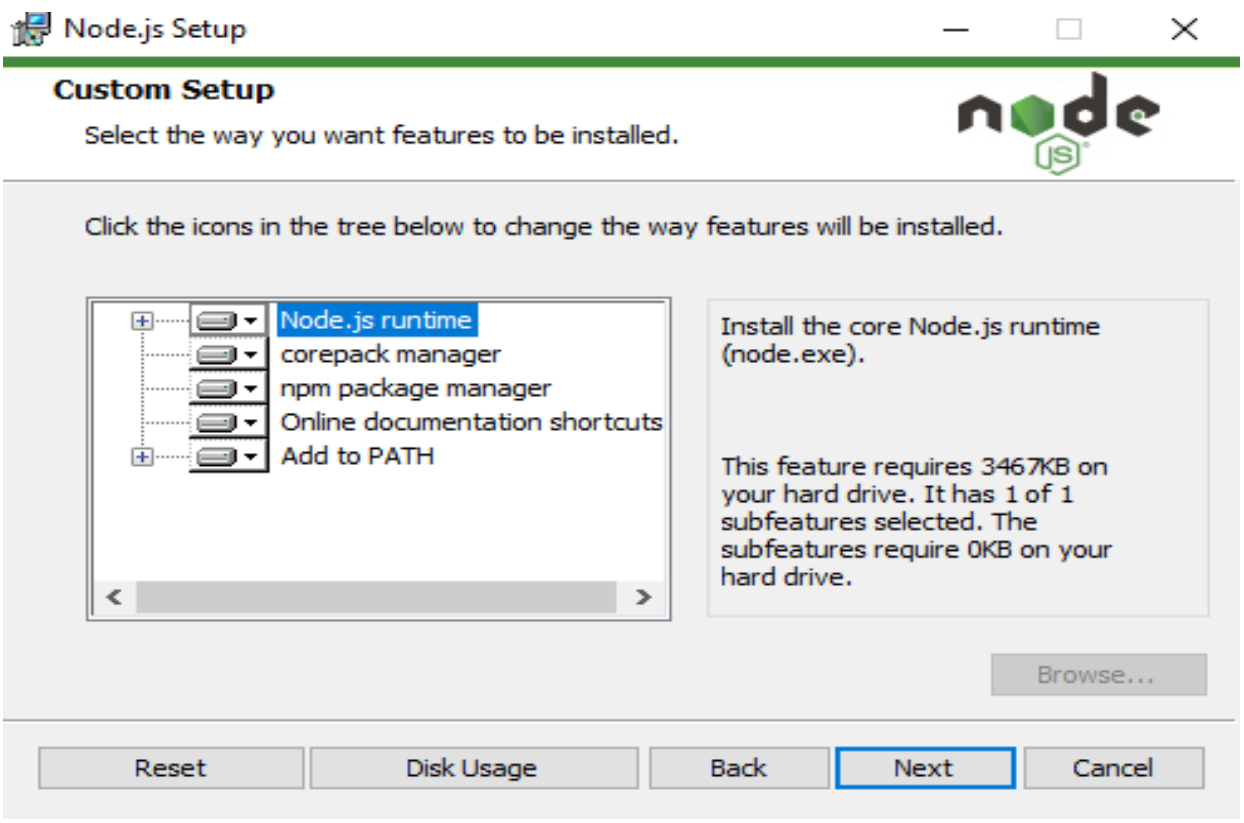

Fig. B.4.2 *Node.js Setup*

B.5 Firebase Database Connection

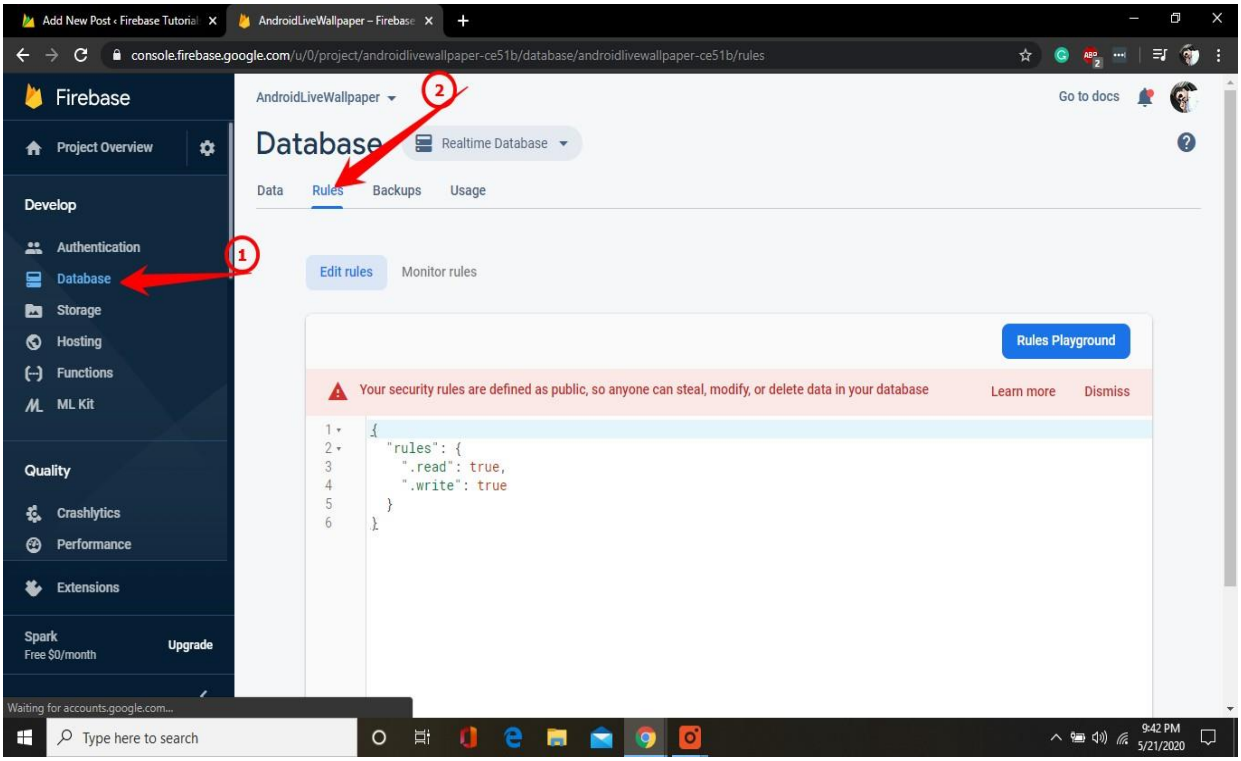

Fig. B.5 *Database creation in Firebase*

APPENDIX – C : SOFTWARE USAGE PROCESS

Using React.js for the front-end, we need to set it up within the project. This typically involves creating a new React app using the Create React App tool.

In the SSPMS project directory, run:

```
npx create-react-app sspms
```

Install the required libraries like react-router-dom and the firebase.

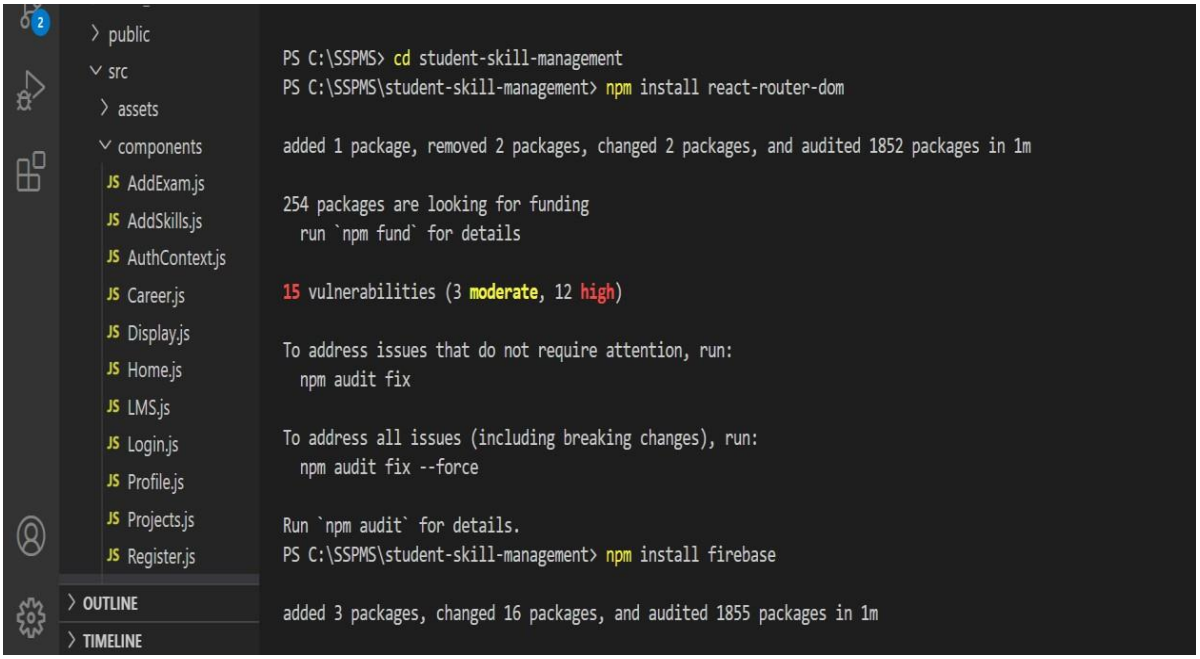

Fig. C.1 *Naviagte to the project folder and install libraries.*

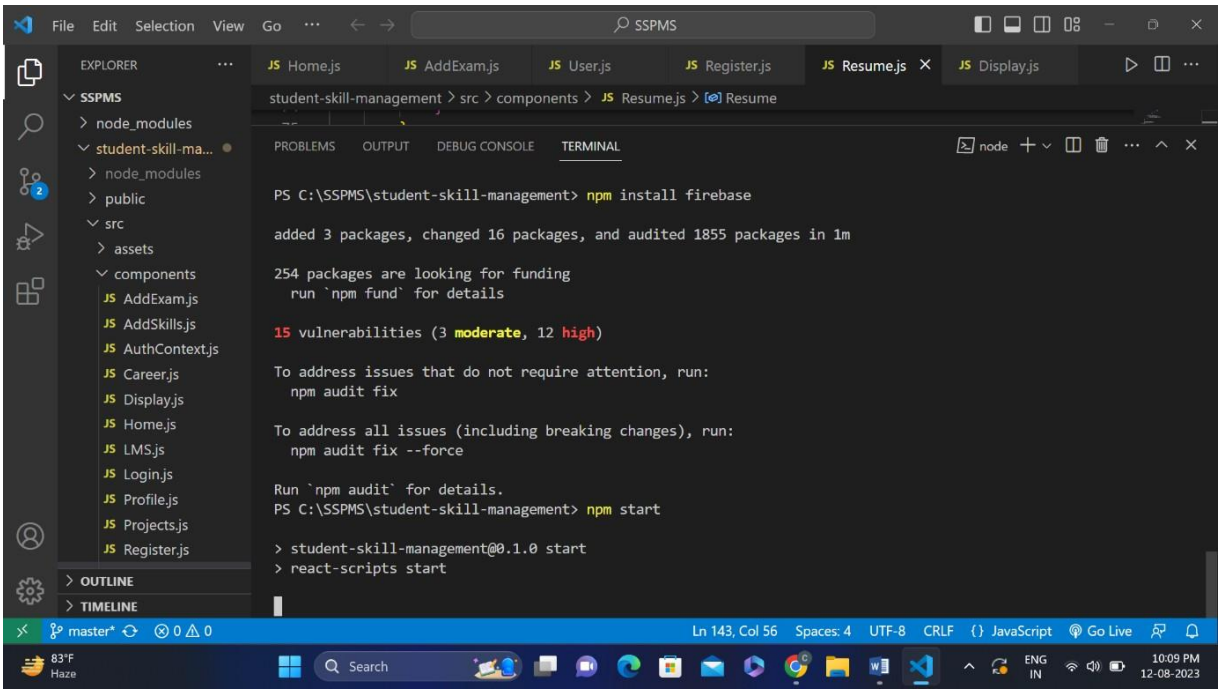

Fig. C.2 *Start the react project development server using the command "npm start"*

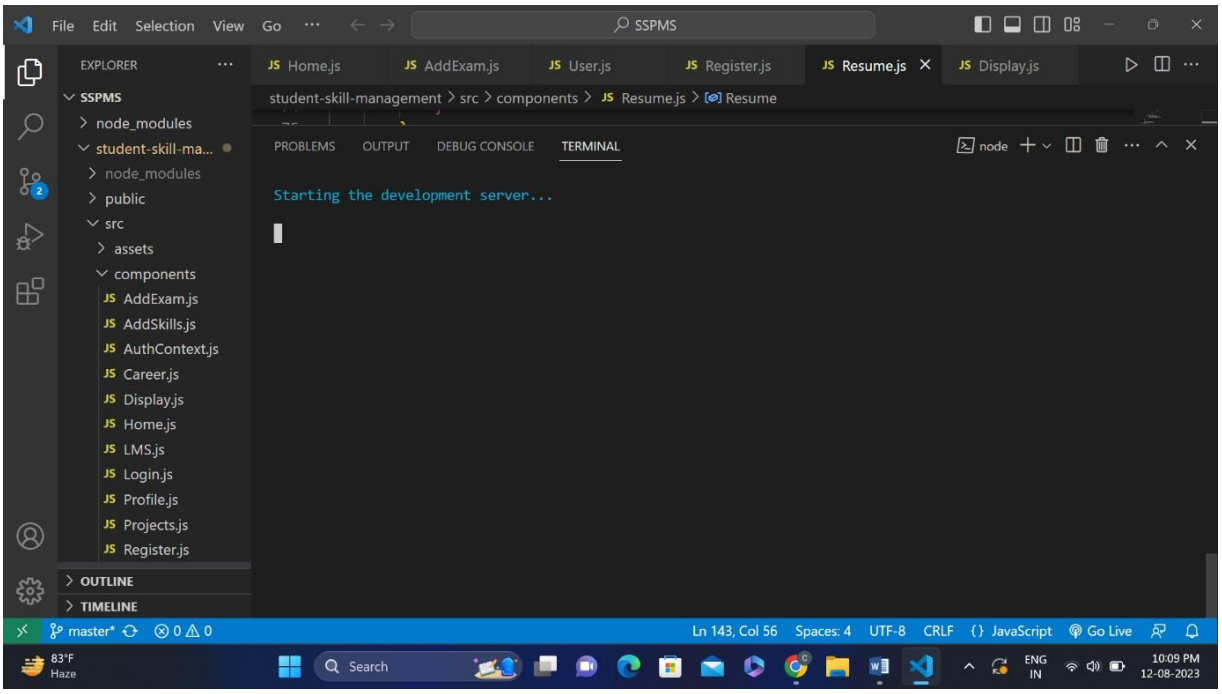

Fig. C.3 *Start of the development server*