



**SRI KRISHNA COLLEGE OF TECHNOLOGY**  
An Autonomous Institution | Accredited by NAAC with 'A' Grade  
Affiliated to Anna University | Approved by AICTE  
**KOVAIPUDUR, COIMBATORE 641042**



## **ADMISSION WEBSITE**

## **A PROJECT REPORT**

*Submitted by*

**Mahisa.P**

**727821TUIT056**

*in partial fulfilment for the award of the degree*

Of

**BACHELOR OF TECHNOLOGY**

**IN**

**INFORMATION TECHNOLOGY**

**JUNE 2023**

**CERTIFICATE**

---

## BONAFIDE CERTIFICATE

Certified that this project report “**Admission Website**” is the Bonafide work of **Mahisa.P** . who carried out the project work under my supervision.

SIGNATURE

**Dr. Siamala Devi S**

**HEAD OF THE DEPARTMENT**

Associate Professor

Department of Information Technology

Sri Krishna College of Technology

Kovaipudur

Coimbatore-42.

SIGNATURE

**Madumidha S**

**SUPERVISOR**

Assistant Professor

Department of Information Technology

Sri Krishna College of Technology

Kovaipudur

Coimbatore-42.

---

Certified that the candidates were examined by us in the Project Work viva-voce examination held on..... at Sri Krishna College of Technology, Coimbatore -641 042.

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## **ACKNOWLEDGEMENT**

---

## ACKNOWLEDGEMENT

First and foremost, we thank the **Almighty** for being our light and for showering his gracious blessings throughout the course of this project.

We are grateful to our beloved Principal **Dr.M.G. Sumithra M.E., Ph.D.** for their tireless and relentless support.

We extend our sincere thanks to our Head of the Department **Dr Siamala Devi S** for her encouragement and inspiration.

We are greatly indebted to our Industry Mentor **Mr.KARAN D** & Project guide **Madumidha S** Assistant Professor, Department of **Information Technology** for theirvaluable guidance and suggestions in all aspects that aided us to ameliorate our skills.

We are thankful to all those who have directly and indirectly extended their help to us in completing this project work successfully.

## **ABSTRACT**

---

## ABSTRACT

With the increasing prominence of e-commerce and the ever-growing demand for convenience, the ice cream industry has embraced online shopping platforms to provide customers with a delightful and hassle-free purchasing experience. This abstract explores the realm of online ice cream shopping and highlights the transformative impact it has had on consumer behavior, market dynamics, and the overall ice cream industry.

The abstract also highlights the challenges and considerations associated with online ice cream shopping. Maintaining the quality and integrity of ice cream during transportation presents logistical hurdles that require careful attention. Adequate packaging, temperature control, and delivery logistics are crucial aspects to ensure the ice cream reaches customers in optimal condition. Furthermore, customer trust and satisfaction rely heavily on reliable delivery networks and efficient customer support systems that promptly address any issues that may arise during the online shopping journey.

Overall, online shopping has revolutionized the ice cream industry by providing an accessible, convenient, and enjoyable purchasing experience for consumers while opening up new growth opportunities for businesses. The abstract concludes by emphasizing the need for continuous innovation and adaptation to evolving customer preferences and technological advancements to ensure the sustainable growth and success of online ice cream shopping in the future.

## TABLE OF CONTENTS

---



## TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	<b>ABSTRACT</b>	<b>IV</b>
	<b>LIST OF FIGURES</b>	<b>VIII</b>
	<b>LIST OF ABBREVIATIONS</b>	<b>IX</b>
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 Problem Statement	2
	1.2 Overview	2
	1.3 Objective	2
<b>2</b>	<b>LITERATURE SURVEY</b>	<b>4</b>
<b>3</b>	<b>ENVIRONMENTAL SETUP</b>	
	3.1 Installation of VS Code	6
	3.2 Installation of Eclipse IDE	7
<b>4</b>	<b>METHODOLOGIES</b>	
	4.1 CRUD Operation Page	11
<b>5</b>	<b>DESIGN AND TECHNOLOGIES</b>	
	5.1 Rest API	15
	5.2 MySQL	16
	5.3 HTML	16
	5.4 CSS	17
	5.5 React	17
<b>6</b>	<b>FUNCTIONAL DESCRIPTION</b>	
	<b>6.1 FRONTEND</b>	
	6.1.1 Home Page	20

	6.1.2 login Page	22
	6.1.3 sign in Page	24
	6.1.4 About Page	30
	6.1.5 contact Page	32
	6.1.6 Dashboard page	34
	6.1.7 List page	35
<b>7</b>	<b>CONCLUSION</b>	<b>39</b>
<b>8</b>	<b>REFERENCES</b>	<b>39</b>

## **LIST OF FIGURES**

---

## LIST OF FIGURES

FIGURE NO	FIGURE NAME
3.1.1	VS Code
3.2.1	Eclipse Download Website
3.2.2	Eclipse Installer
3.2.3	Directory Page
5.1.1	Rest API Packages
5.2.1	MySQL Database
5.3.1	HTML Structure
5.4.1	CSS Structure
5.5.1	JS Structure
5.6.1	React Files
6.1.1	Home Page
6.1.2	Read Page
6.1.3	Update Page
6.1.4	About Page
6.3.1	Database

## **LIST OF ABBREVIATIONS**

---

## LIST OF ABBREVIATIONS

### ACRONYM

### ABBREVIATIONS

JS

JavaScript

MySQL

My Structured Query Language

DOM

Document Object Model

JSX

JavaScript XML







# **CHAPTER 1**

## **INTRODUCTION**

At our institution, we understand that every student is unique, with their own dreams, aspirations, and talents. That's why we're thrilled to introduce you to our admission website, where your journey to academic excellence begins.

Here, we strive to make the application process as smooth and transparent as possible, ensuring that you have all the information and resources you need to embark on this exciting chapter of your life.

Our website is designed to provide comprehensive insights into our institution, its programs, admission requirements, and the vibrant campus community you'll soon be a part of. Whether you're a prospective undergraduate, graduate, or international student, we have tailored resources to guide you through every step of the application process.

From exploring our diverse range of programs to understanding financial aid options, from connecting with faculty members to scheduling campus tours, our admission website serves as your one-stop destination for all things related to joining our academic family.

We believe in fostering a supportive and inclusive environment where students can thrive both academically and personally. Through our website, you'll discover the myriad opportunities for intellectual growth, extracurricular engagement, and community involvement that await you.

Our commitment to excellence extends beyond the classroom, and we're dedicated to empowering each student to reach their full potential. Whether you're pursuing your passion in the sciences, arts, humanities, or beyond, our institution is dedicated to providing you with the tools, support, and guidance you need to succeed.

So, take the first step towards a transformative educational experience by exploring our admission website today. We can't wait to welcome you into our vibrant community and embark on this journey of discovery together.

**Problem Statement:**

Website which helps the students who seeking for good colleges all over the world. In our website, we have many colleges enlisted in our website. In that website, list of colleges can be view

**Overview:**

Immerse yourself in the vibrant campus community through organizations, events, and activities designed to enhance your college experience. Connect with our friendly admissions team for personalized assistance and guidance as you embark on your journey towards academic success.

**Objective:**

Our objective in developing this admission website for higher studies is to create a comprehensive and engaging platform that caters to the needs of prospective students aiming to advance their education and career aspirations. We strive to provide clarity by offering clear and concise information about our institution, including detailed insights into our diverse range of programs, admission requirements, and application procedures.

## **LITERATURE SURVEY**

---

## CHAPTER 2

### LITERATURE SURVEY

#### Key Findings:

1. **User-Friendly Design:** The website features a user-friendly design with intuitive navigation, making it easy for prospective students to find relevant information about programs, admission requirements, and procedures.
2. **Comprehensive Information:** The website provides comprehensive information about the institution, including detailed insights into academic programs, faculty expertise, campus life, and support services available to students.
3. **Interactive Features:** Interactive features such as virtual tours, live chats with admissions representatives, and social media integration facilitate engagement and allow prospective students to connect with the academic community.
4. **Clear Communication:** The website effectively communicates admission requirements, application deadlines, and procedures, enabling prospective students to make informed decisions about their higher education journey.
5. **Accessibility:** The website prioritizes accessibility, ensuring that it is inclusive and accessible to individuals from diverse backgrounds and abilities, thus widening the reach of the institution's offerings.

## ENVIRONMENTAL SETUP

---

## CHAPTER 3

# ENVIRONMENTAL SETUP

### 3.1 Installation of Visual Studio Code

A software package known as an integrated development environment, or IDE, integrates the essential tools required to design and test software. Developers use a range of tools for writing, creating, and testing software code. Examples of development tools include text editors, code libraries, compilers, and test environments. Without an IDE, a developer must pick, utilize, integrate, and manage each of these tools individually. The IDE used for this project is Visual Studio Code. The main advantage of VSC is that it offers a tone of features and code snippets that hasten implementation. Developers can give commands immediately on VSC without needing to open a terminal process thanks to a built-in terminal system on VSC, which makes it easier to check for errors. The fact that VSC comes with a built-in GIT command that enables diff checking, staging files, and committing directly from the editor is another reason to choose it. Additionally, a sizable selection of libraries of extensions and plug-ins that support the coding of the most widely used programming languages in use today are available on the Visual Studio market.

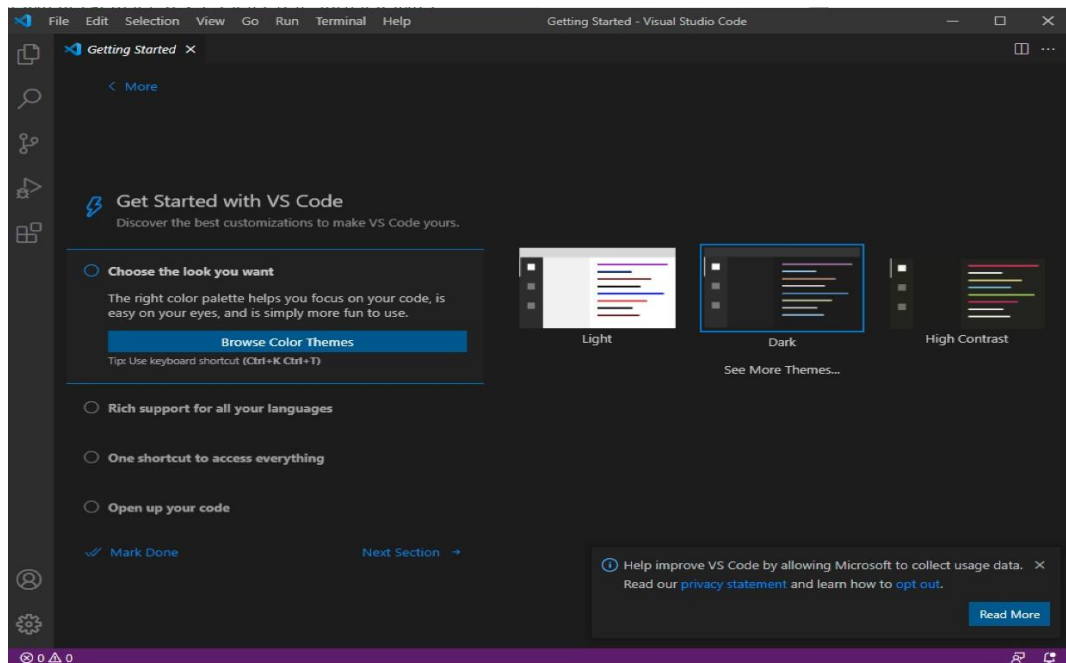
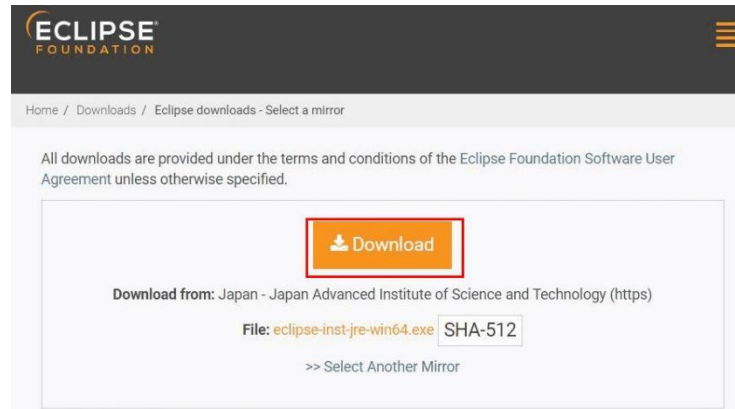


Figure 3.1.1 VS Code

## 3.2 Installation of Eclipse IDE

Download Eclipse from the official website [www.eclipse.org](http://www.eclipse.org). Click on x86\_64 button and click on the Download button and download of Eclipse IDE will begin. After Downloading double click the file.



**Figure 3.2.1 Eclipse Download Website**

Eclipse installer dialog box appear on the screen. Click on Eclipse IDE for Java Developers and click Next.



**Figure 3.2.2 Eclipse Installer**

The directory dialog box will appear here select the folder or path where you want to install your Eclipse IDE. Select the path and Click Install.



**Figure 3.2.3 Directory Page**

Now the Eclipse IDE has installed on your system. Now click on Create New Java Project and start building your Java projects.



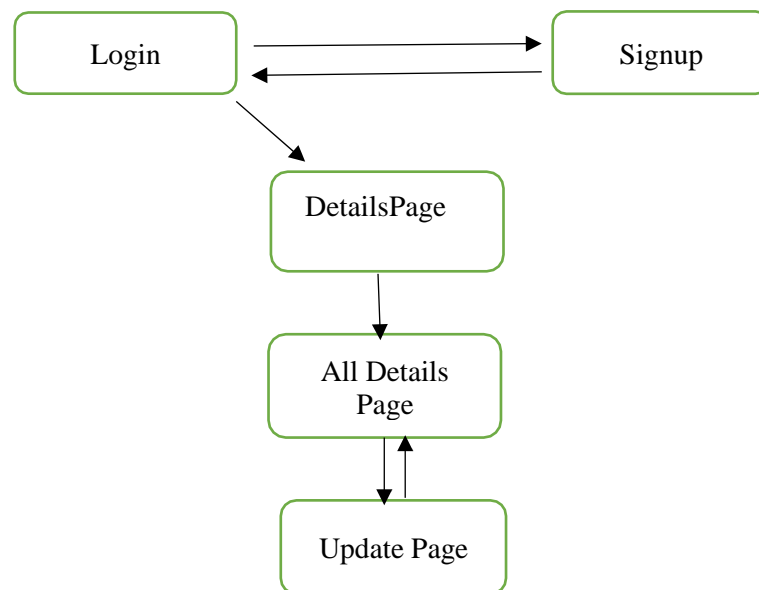
## **METHODOLOGIES**

---

## CHAPTER 4

### METHODOLOGIES

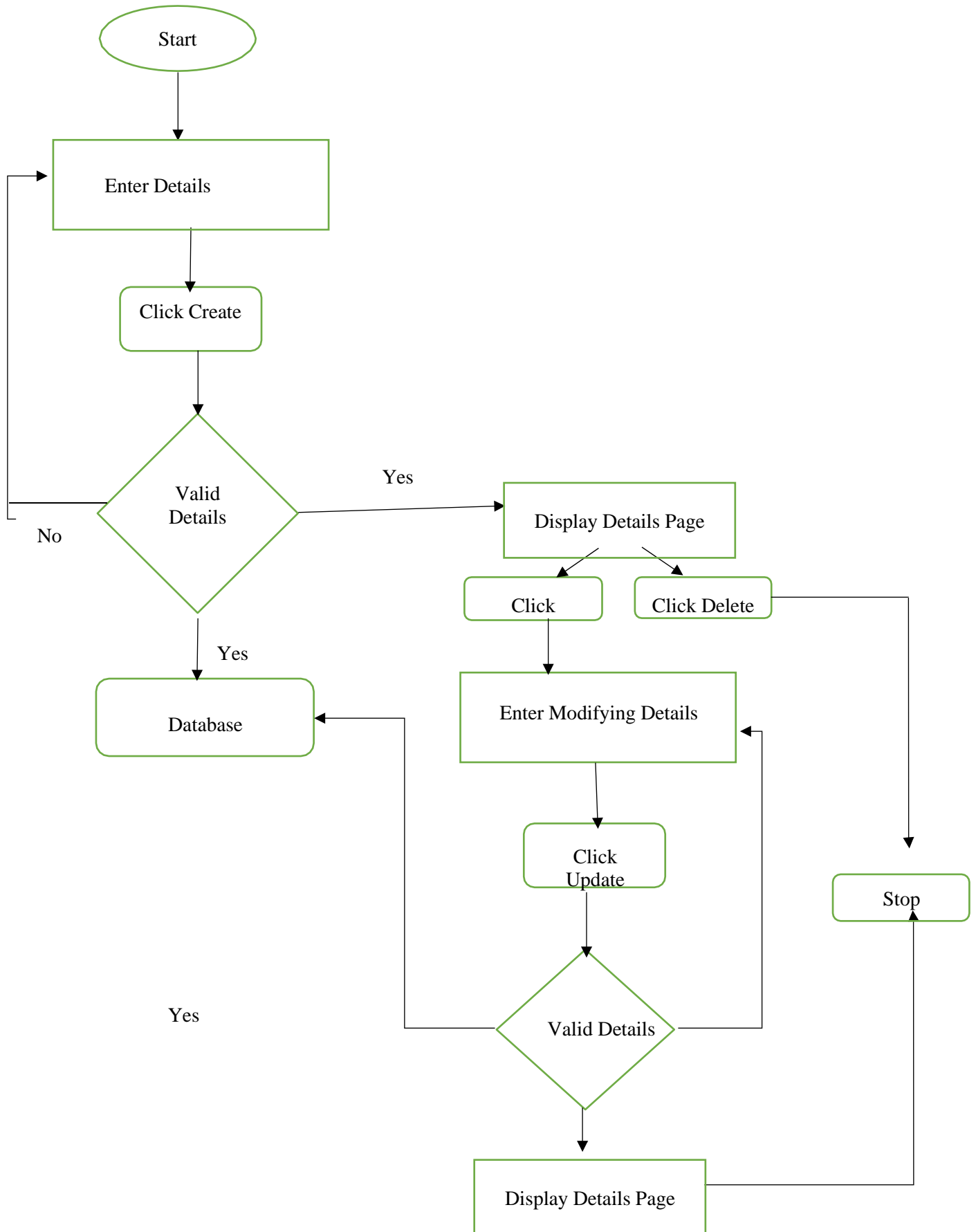
This chapter gives a small description about how our system works.



The above give idea of the flow of the website. First User will Login if he has account if he doesn't he have to go to Signup Page. The Officials or Supervisor will give the details about them and they will their details in the database.



## 4.1 CRUD Operation Flow



above flowchart represents the CRUD operations and its flow of control. At first the user will enter the animal details in the Animal Details Page and the user will click on create button if the details are valid, the details will be stored in the database and then it will move to the Display Details Page if the entered details are invalid it will remain in the same page.

In Display Details Page the details that are entered will be displayed and there will be two buttons named Delete and Update. If the user clicks on Update button the site will route to Update Page and the details that are stored in the database will be displayed and the user will change details as they wish and clicks on Update button if the details are valid, the updated details will be stored in the database or else it will remain in the same page until the user enters the correct details. When the user clicks Delete button the record will be deleted from the frontend and from the database.



## CHAPTER 5

### TOOLS AND TECHNOLOGIES

The Project “Wildlife Management System” uses lots of tools and technologies. Some of them are listed below. They are

- Rest API
- MySQL
- HTML
- CSS
- React

#### 5.1 Rest API

An API is a set of definitions and protocols for building and integrating application software. It's sometimes referred to as a contract between an information provider and an information user—establishing the content required from the consumer (the call) and the content required by the producer (the response). Here we are using tool called Eclipse IDE. The folder structure is as follows:

```

v Wildlife [boot] [devtools]
v src/main/java
  > com.fullstack.Wildlife
  v com.fullstack.Wildlife.Controller
    > AnimalListController.java
    > LoginController.java
    > SignupController.java
  v com.fullstack.Wildlife.Entity
    > AnimalListEntity.java
    > LoginEntity.java
    > SignupEntity.java
  v com.fullstack.Wildlife.Repository
    > AnimalListRepo.java
    > LoginRepo.java
    > SignupRepo.java
  v com.fullstack.Wildlife.Service
    > AnimalListService.java
    > LoginService.java
    > SignupService.java
  > src/main/resources
  > src/test/java
  > JRE System Library [JavaSE-17]
  > Maven Dependencies
  > src
  > target
  > HELP.md
  > mvnw

```

## 5.2 MySQL

MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise application.

```
Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| fullstack |
| fullstack1 |
| hospital_database |
| information_schema |
| mysql |
| performance_schema |
| sys |
| wildlife |
+-----+
8 rows in set (0.03 sec)
```

**Figure 5.2.1 MySQL**

## 5.3 HTML

HTML means Hypertext Markup Language. This language is used in creating web pages. This language also supports other languages such as CSS, PHP, JAVASCRIPT, etc. in creating interactive and responsive pages on the web. HTML5 is just an updated version of the HTML. It supports new features, new attributes, new HTML elements, full CSS3 support, video and audio, 2D/3D graphics that help users and also help web developers to create new features easily on the website. The structure of HTML5 is shown

```
<!DOCTYPE html>
<html>

  <head>
    <title> Title here </title>
  </head>

  <body>
    Web page content goes here.
  </body>

</html>
```

**Figure 5.3.1 HTML Structure**



## 5.4 CSS – CASCADING STYLE SHEET

CSS is simply referred to as Cascading Style Sheets. CSS is used to define styles for web pages, including the design, layout, and variations in the display for different devices and screen sizes. The general structure of CSS

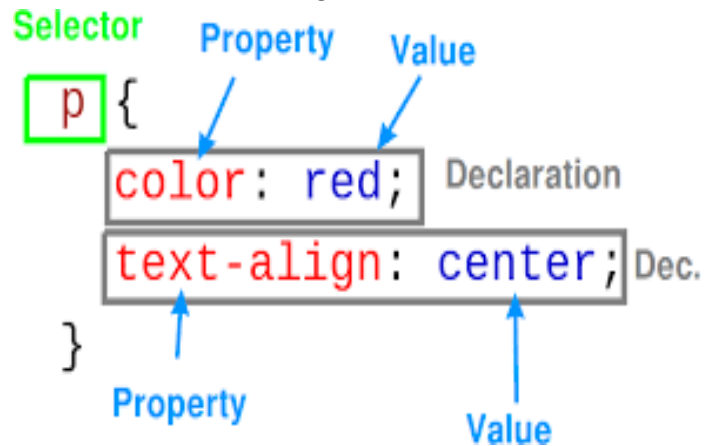
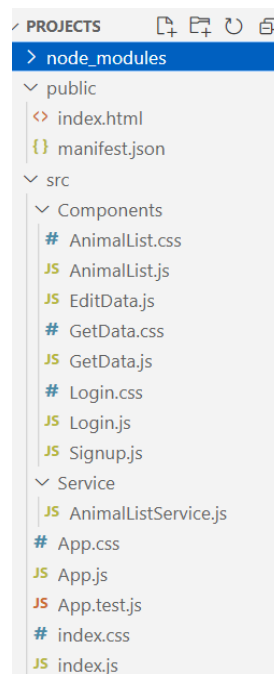


Figure 5.4.1 CSS Structure

## 5.5 React

React is a **JavaScript-based UI development library**. Facebook and an open source developer community run it. Although react is a library rather than a language, it is widely used in web development. The library first appeared in May 2013 and is now one of the most commonly used frontend libraries for web development. At the end of the project our React project folders are:





**FUNCTIONAL DESCRIPTION**

---

## CHAPTER 6

### FUNCTIONAL DESCRIPTION

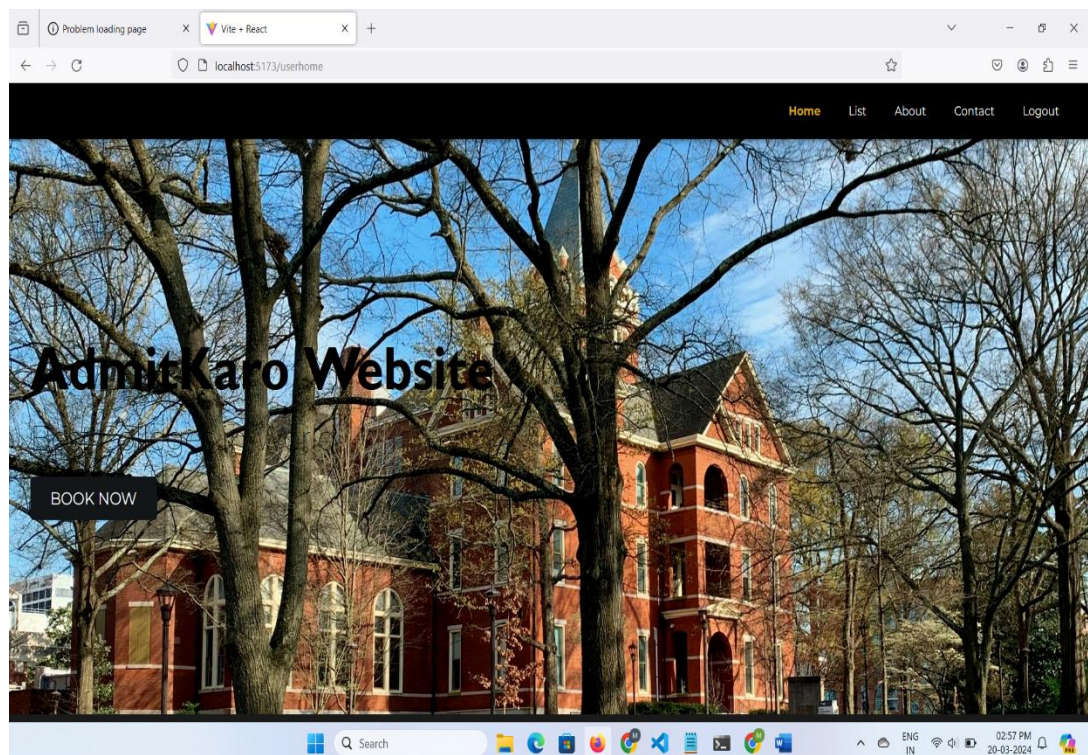
In this chapter the detailed explanation of the project will be seen. How the front-end i.e., the UI works and the backend connection will be seen.

#### 6.1 Frontend

A front-end developer is a type of software developer who specializes in creating and designing the user interface (UI) and user experience (UX) of websites and web applications. The primary responsibility of a front-end developer is to ensure that the visual and interactive aspects of a website or application are user-friendly, aesthetically pleasing, and functionally efficient.

##### 6.1.1 Home Page

The Details Page is a page which allows the officials to enter the information about those who came to their supervision for the first time. The Details page consists of, Gender and Funding sources. The Entry page will be shown below



**6.1.1 Figure Home Page**

```

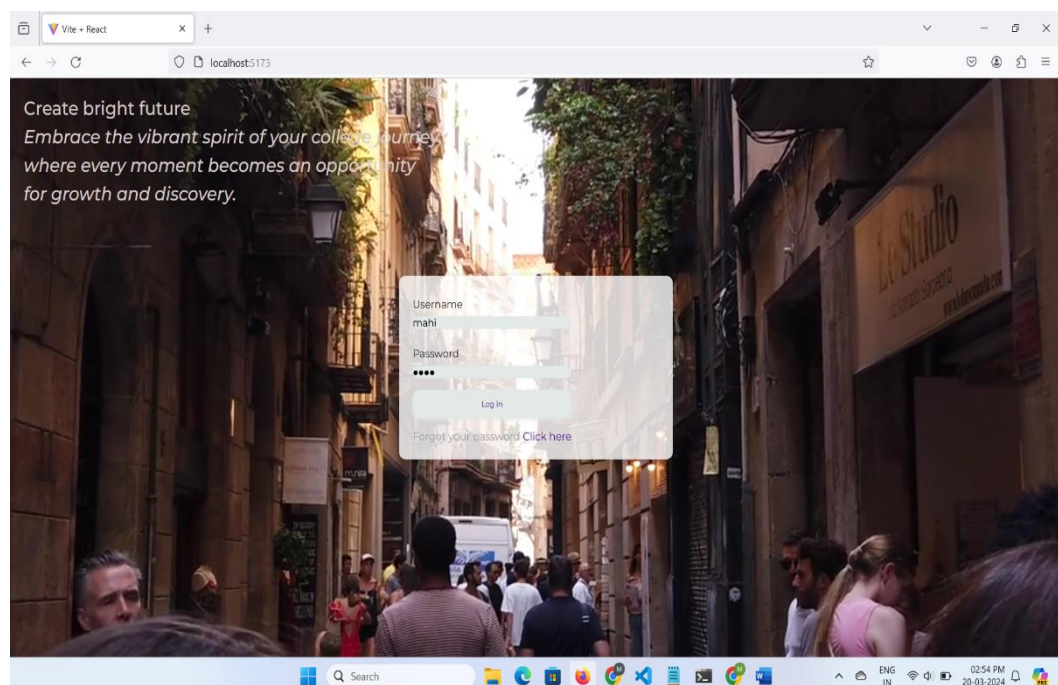
import React from "react";
import Layout from "../Layout/Layout";
import { Link } from "react-router-dom";
import Banner from "../Assets/i1.jpg";
import "../styles/HomeStyles.css";

const Home = () => {
  return (
    <Layout>
      <div className="home" style={{ backgroundImage: `url(${Banner})` }}>
        <div className="headerContainer">
          <h1>AdmitKaro Website</h1>
          <p>Best Colleges in World</p>
          <Link to="/list">
            <button>BOOK NOW</button>
          </Link>
        </div>
      </div>
    </Layout>
  );
};

export default Home;

```

## 6.1.2 Login Page



### 6.1.2 Login Page Figure.js Code:

```
import React, { useState } from 'react';
import { Link } from 'react-router-dom';
import './Login.css';
import '../App.css';
import video from '../LoginAssets/video.mp4';
import logo from '../LoginAssets/logo.png';
import { FaUserAlt, FaLock } from 'react-icons/fa';
import { AiOutlineSwapRight } from 'react-icons/ai';

const Login = () => {
  const [formData, setFormData] = useState({
    username: '',
    password: '',
  });

  const [errors, setErrors] = useState({});

  const handleSubmit = (e) => {
    e.preventDefault();
    const { username, password } = formData;

    // Check if username and password match admin credentials
    if (username === 'admin' && password === 'mahi@123') {
      // Redirect to dashboard
      window.location.href = '/dashboard';
    } else {
      // Check if username and password match registered info
      // Replace these values with actual registered information

    }
  };

  const handleChange = (e) => {
    setFormData({
      ...formData,
      [e.target.id]: e.target.value,
    });
    // Clear errors when user starts typing
    setErrors({});
  };
};
```

```

return (
  <div className='loginPage flex'>
    <div className='container flex'>

      <div className='videoDiv'>
        <video src={ video } autoPlay muted loop></video>
        <div className='textDiv'>
          <h2 className='title'>Create bright future<br/>
            <i>Embrace the vibrant spirit of your college journey,<br/> where every
moment becomes an opportunity <br/>for growth and discovery. </i>
          </h2>
        </div>
        <div className='footerDiv flex'>
          <span className='text'>Don't have any account</span>
          <Link to='/register'>
            <button className='btn'>Sign up</button>
          </Link>
        </div>
      </div>

      <div className='formDiv flex'>
        <div className='headerDiv'>

          </div>

        <form onSubmit={handleSubmit} className='form grid'>
          {errors.invalidCredentials && <span
className='error'>{errors.invalidCredentials}</span>}
          <div className='inputDiv'>
            <label htmlFor='username'>Username</label>
            <div className='input flex'>
              <input
                type='text'
                id='username'
                placeholder='Enter Username'
                value={ formData.username }
                onChange={handleChange}
              />
            </div>
          </div>
          <div className='inputDiv'>
            <label htmlFor='password'>Password</label>
            <div className='input flex'>
              <input
                type='password'
                id='password'

```

```

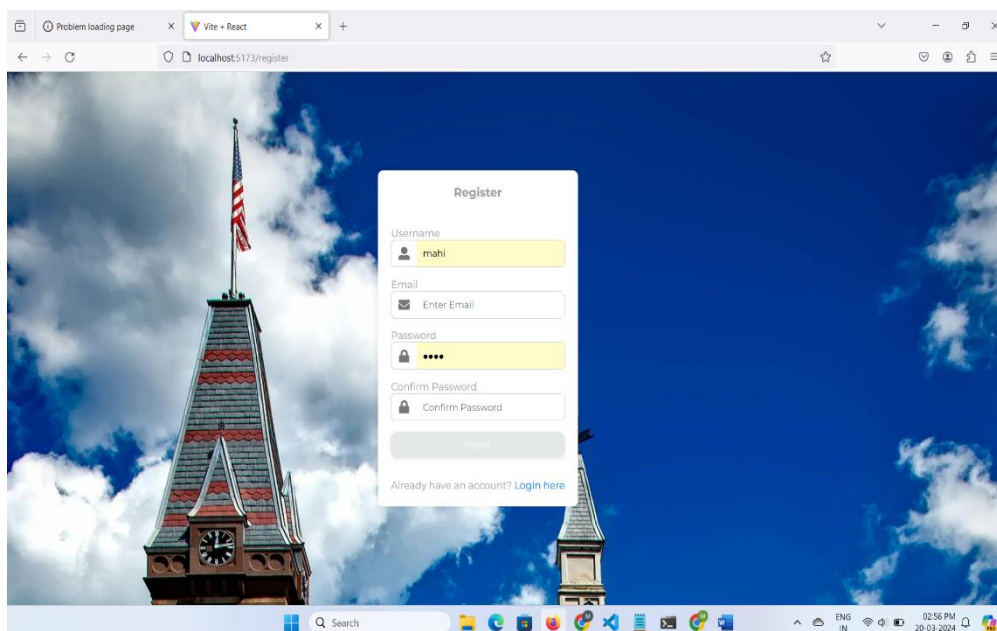
        placeholder='Enter password'
        value={ formData.password}
        onChange={handleChange}
      />
    </div>
  </div>
  <button type='submit' className='btn flex'>
    <span><Link to='/userhome'>Log in</Link></span>
  </button>
  <span className='forgot password'>
    Forgot your password <Link to='/register'>Click here</Link>
  </span>
</form>
</div>
</div>
</div>
);
};

export default Login;

```

### 6.1.3 Register Page

This page enables the user to register the user which have been already entered by the user. This page will be able, and price which has been already entered by the user during the first time order.



**Figure 6.1.3 Register Page.js Code:**



```

import React, { useState } from 'react';

import './Register.css' // You can define your CSS styles here

import { Link } from 'react-router-dom';

import { FaUserAlt, FaLock, FaEnvelope } from 'react-icons/fa';

import backgroundImage from '../Assets/i2.jpg'; // Import your image

const Register = () => {

  const [formData, setFormData] = useState({

    username: "",

    email: "",

    password: "",

    confirmPassword: "",

  });

  const [errors, setErrors] = useState({ });

  const handleSubmit = (e) => {

    e.preventDefault();

    // Validation logic

    const newErrors = { };

    if (!formData.username.trim()) {

      newErrors.username = 'Username is required';

    }

    if (!formData.email.trim()) {

      newErrors.email = 'Email is required';

    } else if (!/^$+@\$+.\$+/.test(formData.email)) {

      newErrors.email = 'Invalid email address';

    }

  }

```

```

if (!formData.password.trim()) {
  newErrors.password = 'Password is required';
} else if (formData.password.length < 6) {
  newErrors.password = 'Password must be at least 6 characters long';
}

if (formData.password !== formData.confirmPassword) {
  newErrors.confirmPassword = 'Passwords do not match';
}

if (Object.keys(newErrors).length > 0) {
  setErrors(newErrors);
} else {
  // If no errors, submit form
  // Example: Call an API for registration
  console.log('Form submitted:', formData);
  // Redirect to login page after successful registration
  // Use Link for redirection
}
};

const handleChange = (e) => {
  setFormData({
    ...formData,
    [e.target.id]: e.target.value,
  });
  // Clear errors when user starts typing
  setErrors({
    ...errors,
    [e.target.id]: '',
  });
};

```

```

};

return (
  <div className='registerPages flex' style={{ backgroundImage:
`url(${backgroundImage})` }}>
    <div className='containers flex'>
      <div className='formDivs'>
        <div className='headerDivs'>
          <h3>Register</h3>
          <br/>
        </div>

        <form onSubmit={handleSubmit} className='form'>
          </* Display validation errors */>
          {errors.username && <span
className='error'>{errors.username}</span>}

          <div className='inputDivs'>
            <label htmlFor='username'>Username</label>
            <div className='inputs flex'>
              <FaUserAlt className='icon' />
              <input
                type='text'
                id='username'
                placeholder='Enter Username'
                value={formData.username}
                onChange={handleChange}
              />
            </div>
          </div>

          {errors.email && <span className='error'>{errors.email}</span>}

```

```

<div className='inputDivs'>

  <label htmlFor='email'>Email</label>

  <div className='inputs flex'>

    <FaEnvelope className='icon' />

    <input
      type='email'
      id='email'
      placeholder='Enter Email'
      value={formData.email}
      onChange={handleChange}
    />

  </div>

</div>

{errors.password && <span
  className='error'>{errors.password}</span>}

<div className='inputDivs'>

  <label htmlFor='password'>Password</label>

  <div className='inputs flex'>

    <FaLock className='icon' />

    <input
      type='password'
      id='password'
      placeholder='Enter Password'
      value={formData.password}
      onChange={handleChange}
    />

  </div>

</div>

{errors.confirmPassword && <span
  className='error'>{errors.confirmPassword}</span>}

```

```

    <div className='inputDivs'>

      <label htmlFor='confirmPassword'>Confirm Password</label>

      <div className='inputs flex'>

        <FaLock className='icon' />

        <input

          type='password'

          id='confirmPassword'

          placeholder='Confirm Password'

          value={formData.confirmPassword}

          onChange={handleChange}

        />

      </div>

    </div>

    <button type='submit' className='btn'>

      Register

    </button>

    <p className='login-link'>

      Already have an account? <Link to='/'>Login here</Link>

    </p>

  </form>

</div>

</div>

</div>

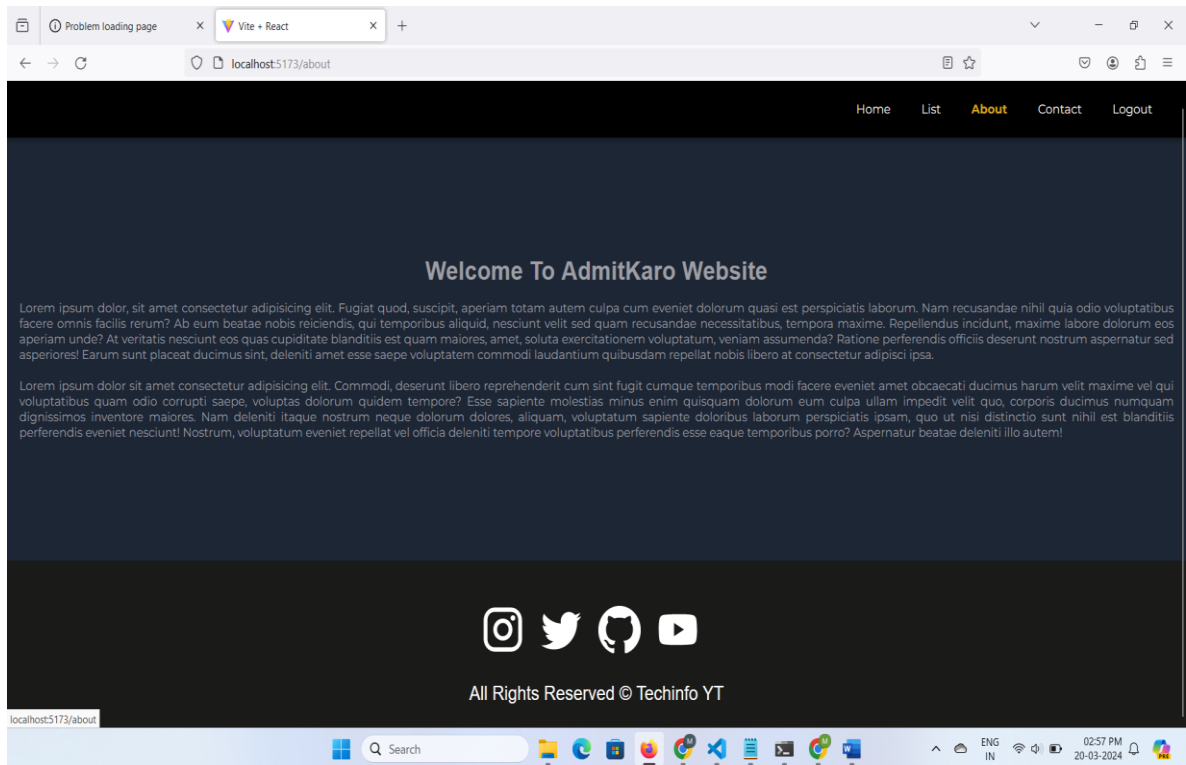
);

};

export default Register;

```

#### 6.1.4 ABOUT PAGE



**Figure 6.1.4 About Page.js Code:**

```
import React from "react";
import Layout from "../Layout/Layout";
import { Box, Typography } from "@mui/material";
```

```
const About = () => {
  return (
    <Layout>
      <Box
        sx={{
          my: 15,
          textAlign: "center",
          p: 2,
          "& h4": {
            fontWeight: "bold",
            my: 2,
            fontSize: "2rem",
          },
          "& p": {
            textAlign: "justify",
          },
          "@media (max-width:600px)": {
            mt: 0,
            "& h4 ": {
```

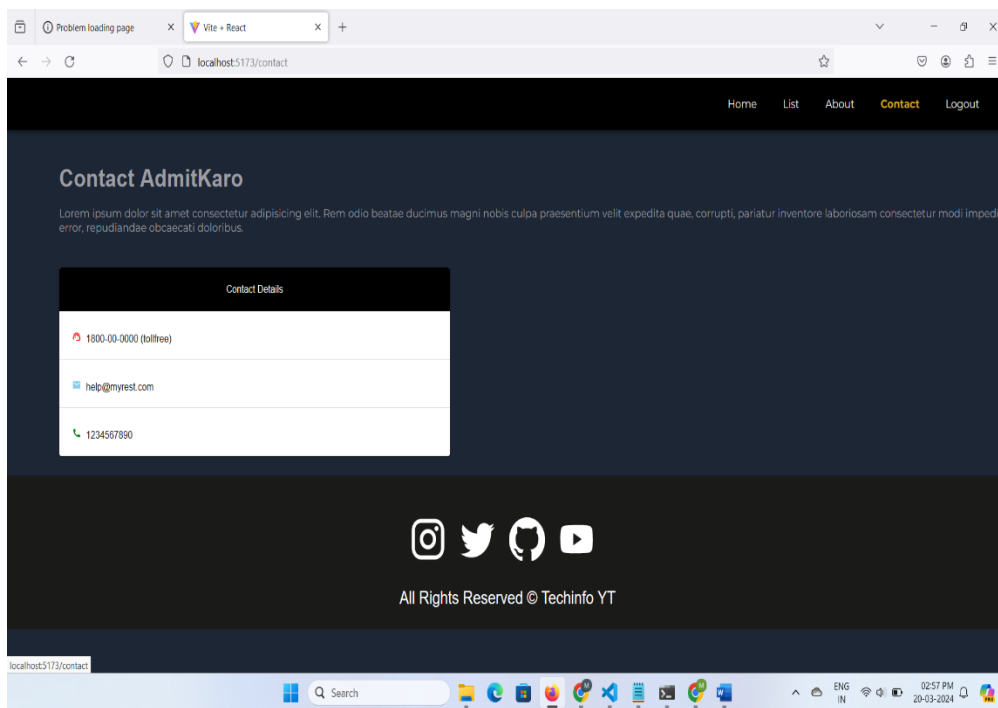
```

        fontSize: "1.5rem",
    },
},
}}
>
<Typography variant="h4">Welcome To AdmitKaro Website</Typography>
<p>
    Lorem ipsum dolor, sit amet consectetur adipisicing elit. Fugiat quod,
    suscipit, aperiam totam autem culpa cum eveniet dolorum quasi est
    perspiciatis laborum. Nam recusandae nihil quia odio voluptatibus
    facere omnis facilis rerum? Ab eum beatae nobis reiciendis, qui
    temporibus aliquid, nesciunt velit sed quam recusandae necessitatibus,
    tempora maxime. Repellendus incidunt, maxime labore dolorum eos
    aperiam unde? At veritatis nesciunt eos quas cupiditate blanditiis est
    quam maiores, amet, soluta exercitationem voluptatum, veniam
    assumenda? Ratione perferendis officiis deserunt nostrum aspernatur
    sed asperiores! Earum sunt placeat ducimus sint, deleniti amet esse
    saepe voluptatem commodi laudantium quibusdam repellat nobis libero at
    consectetur adipisci ipsa.
</p>
<br />
<p>
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Commodi,
    deserunt libero reprehenderit cum sint fugit cumque temporibus modi
    facere eveniet amet obcaecati ducimus harum velit maxime vel qui
    voluptatibus quam odio corrupti saepe, voluptas dolorum quidem
    tempore? Esse sapiente molestias minus enim quisquam dolorum eum culpa
    ullam impedit velit quo, corporis ducimus numquam dignissimos
    inventore maiores. Nam deleniti itaque nostrum neque dolorum dolores,
    aliquam, voluptatum sapiente doloribus laborum perspiciatis ipsam, quo
    ut nisi distinctio sunt nihil est blanditiis perferendis eveniet
    nesciunt! Nostrum, voluptatum eveniet repellat vel officia deleniti
    tempore voluptatibus perferendis esse eaque temporibus porro?
    Aspernatur beatae deleniti illo autem!
</p>
</Box>
</Layout>
);
};

export default About;

```

## 6.1.5 CONTACT PAGE



```
import React from "react";
import Layout from "../Layout/Layout";
import SupportAgentIcon from "@mui/icons-material/SupportAgent";
import MailIcon from "@mui/icons-material/Mail";
import CallIcon from "@mui/icons-material/Call";
import {
  Box,
  Paper,
  Table,
  TableBody,
  TableCell,
  TableContainer,
  TableRow,
  Typography,
} from "@mui/material";

const Contact = () => {
  return (
    <Layout>
      <Box sx={{ my: 5, ml: 10, "& h4": { fontWeight: "bold", mb: 2 } }}>
        <Typography variant="h4">Contact AdmitKaro</Typography>
        <p>
          Lorem ipsum dolor sit amet consectetur adipisicing elit. Rem odio

```



beatae ducimus magni nobis culpa praesentium velit expedita quae,  
 corrupti, pariaturo inventore laboriosam consecetur modi impedit  
 error, repudiandae obcaecati doloribus.

```

</p>
</Box>
<Box
  sx={{
    m: 3,
    width: "600px",
    ml: 10,
    "@media (max-width:600px)": {
      width: "300px",
    },
  }}
>
<TableContainer component={Paper}>
  <Table aria-label="contact table">
    <TableHead>
      <TableRow>
        <TableCell
          sx={{ bgcolor: "black", color: "white" }}
          align="center"
        >
          Contact Details
        </TableCell>
      </TableRow>
    </TableHead>
    <TableBody>
      <TableRow>
        <TableCell>
          <SupportAgentIcon sx={{ color: "red", pt: 1 }} /> 1800-00-0000
            (tollfree)
        </TableCell>
      </TableRow>
      <TableRow>
        <TableCell>
          <MailIcon sx={{ color: "skyblue", pt: 1 }} /> help@myrest.com
        </TableCell>
      </TableRow>
      <TableRow>
        <TableCell>
          <CallIcon sx={{ color: "green", pt: 1 }} /> 1234567890
        </TableCell>
      </TableRow>
    </TableBody>
  </Table>
</TableContainer>

```

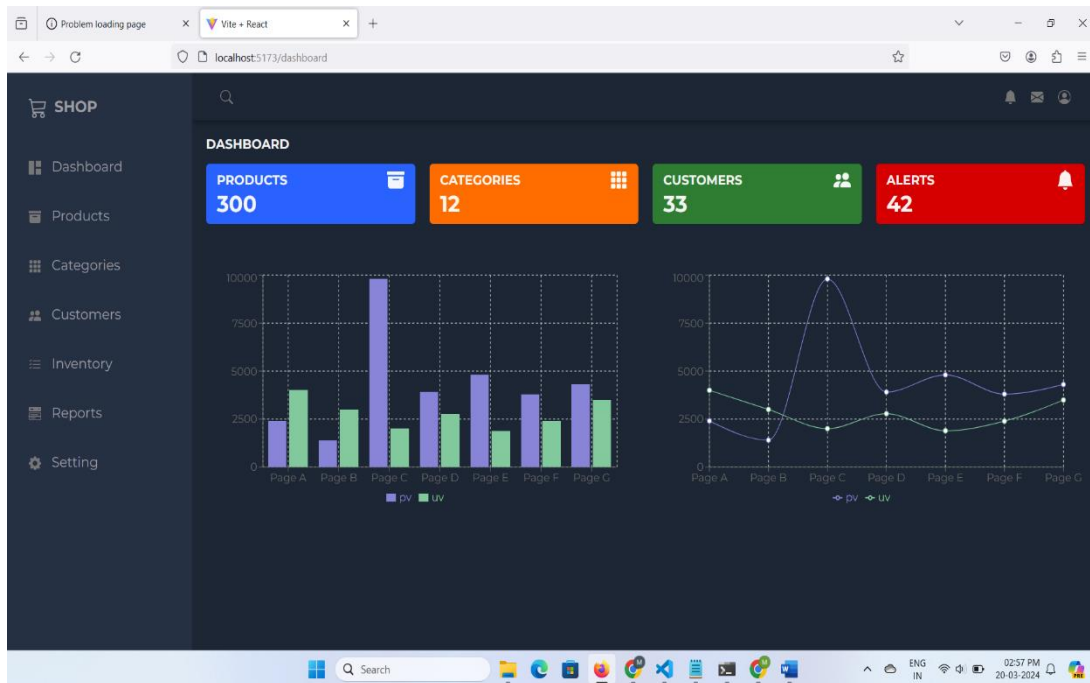
```

    </Table>
  </TableContainer>
</Box>
</Layout>
);
};

```

```
export default Contact;
```

## 6.1.6 DASHBOARD PAGE



```

import React, { useState } from 'react';
import Header from '../Header/Header';
import Sidebar from '../Sidebar/Sidebar';
import Home from '../Home/Home';

```

```

const Dashboard = () => {
  const [openSidebarToggle, setOpenSidebarToggle] = useState(false);

```

```

  const OpenSidebar = () => {
    setOpenSidebarToggle(!openSidebarToggle);
  };

```

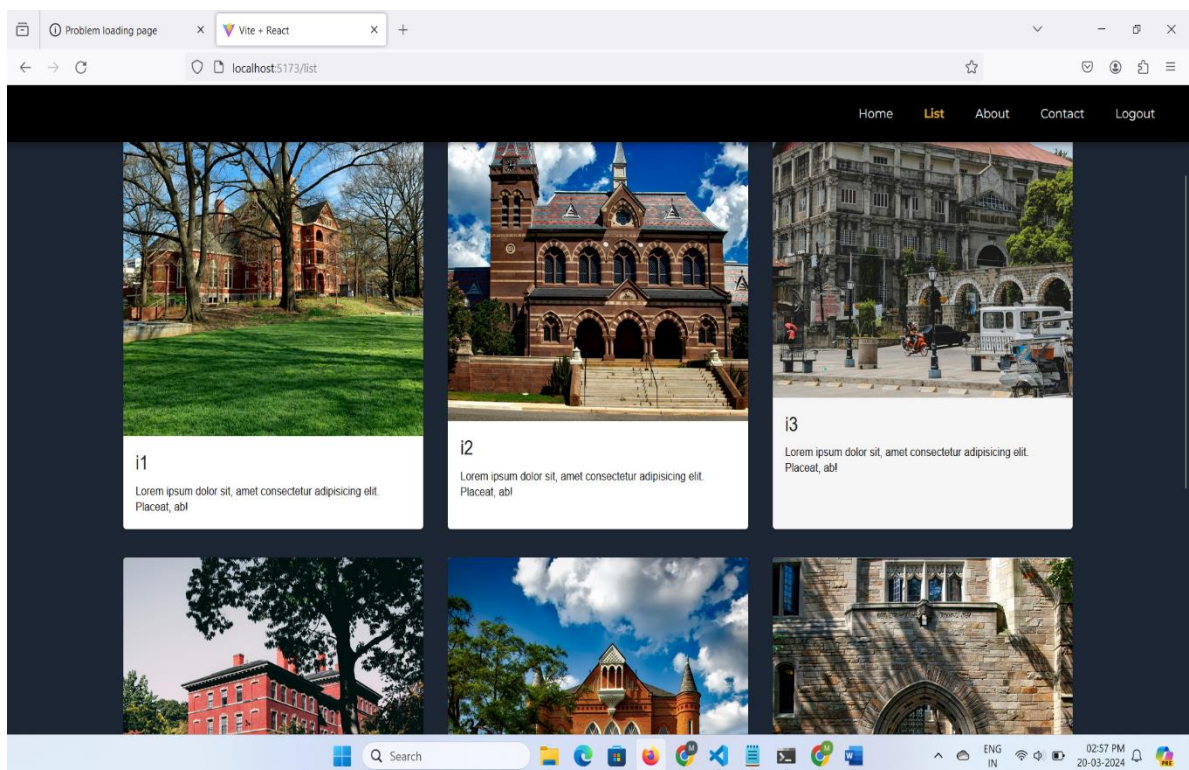
```

return (
  <div className='grid-container'>
    <Header OpenSidebar={OpenSidebar} />
    <Sidebar openSidebarToggle={openSidebarToggle} OpenSidebar={OpenSidebar} />
    <Home />
  </div>
);
};

```

export default Dashboard;

## 6.1.7 LIST PAGE



```

import React from "react";
import { ListC } from "../data/data";
import Layout from "../Layout/Layout";
import {
  Box,
  Card,
  CardActionArea,
  CardContent,
  CardMedia,
  Typography,
} from "@mui/material";

```

```

const List = () => {
  return (
    <Layout>

```

```

<Box sx={{ display: "flex", flexWrap: "wrap", justifyContent: "center" }}>
  {ListC.map((list) => (
    <Card sx={{ maxWidth: "390px", display: "flex", m: 2 }}>
      <CardActionArea>
        <CardMedia
          sx={{ minHeight: "400px" }}
          component={"img"}
          src={list.image}
          alt={list.name}
        />
        <CardContent>
          <Typography variant="h5" gutterBottom component={"div"}>
            {list.name}
          </Typography>
          <Typography variant="body2">{list.description}</Typography>
        </CardContent>
      </CardActionArea>
    </Card>
  ))}
</Box>
</Layout>
);
};

export default List;

```



---

## CONCLUSION

## **CHAPTER 7**

### **CONCLUSION**

In conclusion, the findings from the admission website for higher studies underscore its effectiveness in serving as a comprehensive and user-centric platform for prospective students. Through its user-friendly design, the website ensures easy navigation and access to comprehensive information about academic programs, admission requirements, and support services. The inclusion of interactive features fosters engagement and allows prospective students to connect with the academic community, while clear communication of admission procedures enables informed decision-making. Additionally, the website's commitment to accessibility ensures inclusivity for individuals from diverse backgrounds and abilities, further widening the institution's reach. By effectively promoting academic excellence and providing supportive resources, the website empowers prospective students to navigate the admission process successfully and embark on their higher education journey with confidence. Overall, the admission website for higher studies stands as a testament to the institution's commitment to providing a transformative educational experience and nurturing the next generation of leaders and innovators.

#### **Future Scope:**

In the future, the admission website for higher studies holds immense potential for further enhancements and advancements to better cater to the needs of prospective students and adapt to evolving trends in technology and education. One key area of future development is the implementation of personalized content and experiences, leveraging user preferences and interests to create tailored interactions that enhance engagement and satisfaction. Additionally, the integration of advanced data analytics tools will enable deeper insights into user behavior and preferences, allowing for continuous optimization of the website's effectiveness. Emerging technologies such as virtual reality (VR) and augmented reality (AR) present opportunities to offer immersive campus experiences, virtual classrooms, and

interactive learning environments, providing prospective students with a more realistic preview of campus life and academic programs. Furthermore, the introduction of AI-powered chatbots and virtual assistants can enhance support services, providing instant assistance and guidance to prospective students throughout the application process. Continued emphasis on accessibility features will ensure equal access to information and resources for users with disabilities, while community engagement platforms and multilingual support will foster inclusivity and connect prospective students from diverse backgrounds. Finally, strengthening integration with social media platforms will facilitate seamless sharing of content and enhance outreach efforts, ultimately elevating the prospective student experience and solidifying the institution's reputation as a leader in higher education.



## REFERENCES

---

## REFERENCES

- [1] All Basics [www.w3school.com](http://www.w3school.com)
- [2] The basic diagram of The HTML/HTML5/24/2018 44  
<http://www.modelrumahminimalis.co/htmlmenutemplatesfreedownload/69033.html/top-result-html-menu-templatesfreedownloadinspirationalgenerous-menu-template-html-images-entrylevelresumetemplates-picture-2017-hdj5>
- [3] [www.javatpoint.com](http://www.javatpoint.com)
- [4] Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures.
- [5] Richardson, L., & Amundsen, M. (2013). RESTful Web APIs: Services for a Changing World. O'Reilly Media..
- [6] REST API Tutorial (MDN Web Docs): [https://developer.mozilla.org/en-](https://developer.mozilla.org/en-US/docs/Web/API/REST_API_Tutorial)
- [7] React Documentation: <https://reactjs.org/docs/getting-started.html>
- [8] React Tutorial (MDN Web Docs): [https://developer.mozilla.org/enUS/docs/Learn/Tools\\_and\\_testing/Cl](https://developer.mozilla.org/enUS/docs/Learn/Tools_and_testing/Client)  
ient.

