A Mini Project report on

MyFolio – A Resume Builder

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree

Bachelor of Technology

in

Computer Science and Engineering (AI & ML)

Submitted by

B. NARAYAN SINGH
(20H51A6605)

G. RAHUL
(20H51A66A6)

P. SRITEJA
(20H51A66B3)

Under the esteemed guidance of

Mrs. A. Deepika
Assistant Professor, Dept of CSE (AI&ML)

Department of Computer Science and Engineering (AI & ML)

CMR COLLEGE OF ENGINEERING & TECHNOLOGY
(Autonomous)
(NAAC Accredited with 'A+' Grade & NBA Accredited)
(Approved by AICTE, Permanently Affiliated to JNTU Hyderabad)
KANDLAKOYA, MEDCHAL ROAD, HYDERABAD-501401
2020- 2024

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI & ML)

CERTIFICATE

This is to certify that the Mini Project report entitled "**MyFolio – A Resume Builder**" being submitted by **B. Narayan Singh (20H51A6605)**, **G. Rahul (20H51A66A6)**, **P. Sriteja (20H51A66B3)** in partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering (AI&ML) is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree.

Mrs. A Deepika
Assistant Professor
Dept of CSE (AI & ML)

Dr. P. Sruthi
Associate Professor and HOD
Dept of CSE (AI & ML)

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

We are grateful to **Mrs. A. Deepika**, Assistant Professor, Department of Computer Science and Engineering (AI&ML) for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank **Dr. P. Sruthi**, Head of the Department of Computer Science and Engineering (AI&ML), CMR College of Engineering and Technology, who is the major driving forces to complete my project work successfully.

We are very grateful to **Dr. Vijaya Kumar Koppula**, Dean-Academic, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Dr. V A Narayana**, Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the Teaching & Non- teaching staff of Department of Computer Science and Engineering (AI&ML) for their co-operation.

We express our sincere thanks to **Mr. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care.

Finally, we extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

<div align="right">

B. Narayan Singh   20H51A6605
G. Rahul   20H51A66A6
P. Sriteja   20H51A66B3

</div>

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

In today's job market, the resume has become the number one requirement potential employer's request. A resume is summary of your qualifications, skills, and achievements. Hence it plays a vital role and should be built carefully. Creating such structured and professional resume is a difficult task and consumes too much time. MyFolio, a resume builder is one such web page which simplifies the task of creating a resume for individuals. The system is flexible to be used and reduces the need of thinking and designing an appropriate resume according to qualifications. It is developed to provide an easy means for creating a professional looking resume. Individuals just have to fill up a form that specifies questions from all required fields such as personal questions, educational, qualities, interest, skills and so on. The answers provided by the users are stored and the system automatically generates a well-structured resume.

CHAPTER 1
INTRODUCTION

1. INTRODUCTION

1.1 Literature Review

A resume describes the qualifications and what makes a person unique. To stand out among other applicants, one need a resume that markets their strengths and match for the job. A great resume grabs the attention of employers and recruiters, sells strongest skills and accomplishments of a person and most importantly it gets a job interview. The current system is a manual method in which each task is repeated several times. All of the system's activities that could be automated are done manually. Because all tasks are completed manually, a large workforce is required. In personal information, candidates used to write their name, address, and phone number, which are the fields that applicants used to specify in their resumes. In the past, applicants would include their personal opinions in their resumes. Applicants used to give both their early and recent educational details at that time. Because there were no guidelines defined for the appropriate length of resumes at the time, applicants used to create either too short or too long resumes.

Rinki Tyagi et al solved the problem of manual creation of resumes. Long back resumes were created manually by using MS Word and the format at that time was quite different. These are the fields that applicants used to mention in their resumes, in personal information applicants used to write their name, address, and phone number.. Applicants used to create either too short or too long resumes because at that time there were no standards set for the ideal length of resumes. They proposed an Android application that generates formal format resumes.

Bharti Kungwani et al presented a web application - "Analytical Resume Builder" which provides precise knowledge and analyzed stats. In addition, it provides resumes of graduates i.e., placed students. So, the users will get more clarity on their resume. Furthermore, it also informs about upcoming company drives and campus placements. The application uses statistical analysis by collecting quantitative data and then represents it in a visual format. After the successful building of the resume user can export the file in PDF format.

Ref:

https://www.researchgate.net/publication/313789115_The_Resume_Research_Literature_Where_Have_We_Been_and_Where_Should_We_Go_Next

1.2 Problem Statement

Many job seekers and students struggle with creating an effective and professional resume that effectively showcases their skills and experiences to potential employers. As a result, they may have a difficult time standing out in the job market and securing job interviews. The goal of the resume builder project is to develop a tool that helps job seekers easily create a personalized, professional-grade resume that highlights their strengths and qualifications. The resume builder should be user-friendly and allow users to easily input and organize their information, choose from a variety of templates and formats, and download or print their finished resume.

The better resume builder should meet the following requirements:

- Allow users to input and organize their personal information, work experience, education, and skills.
- Offer a variety of customizable templates and formats to choose from.
- Allow users to download or print their finished resume in a variety of file formats (e.g. PDF, Word).
- Be intuitive and easy to use for users of all skill levels.

1.3 Research Objective

The research objectives of the resume builder project could include:

- Determine the most common challenges and frustrations that job seekers face when creating a resume.
- Identify the key features and functionality that job seekers want and need in a resume builder tool.
- Assess the effectiveness of different templates and formats in attracting the attention of potential employers.
- Evaluate the usability and user experience of the resume builder tool.

Overall, the research objectives of the resume builder project should be focused on understanding the needs and preferences of job seekers and developing a tool that meets those needs in the most effective and user-friendly way possible.

1.4 Project Scope and Limitations

Scope:

The scope of the resume builder project includes the development of a tool that allows job seekers and students to easily create a personalized, professional-grade resume. This includes the following key features and functionality:

- Input and organization of personal information, work experience, education, and skills.
- A variety of customizable templates and formats to choose from.
- The ability to download or print the finished resume in a variety of file formats (e.g., PDF, Word).

Limitations:

- The resume builder tool will not provide resume writing services or perform editing or proofreading on behalf of the user. It is the responsibility of the user to ensure the accuracy and effectiveness of their resume content.
- The tool may not be able to accommodate all types of work experience or education, such as unconventional or international experiences. It is the responsibility of the user to determine the appropriateness of the tool for their specific needs.
- The tool may not be able to generate resumes that meet the specific requirements or preferences of all potential employers. It is the responsibility of the user to customize and tailor their resume to fit the requirements of individual job applications.

CHAPTER 2
BACKGROUND WORK

2. BACKGROUND WORK

2.1 Existing Systems

2.1.1 Introduction

Similarly, to Resume, there are websites that generates resume they are resume io, Canva, zety, job seeker etc. as a person looking for work and applying for positions you need to ensure if you decide to use a resume builder that you choose wisely. Therefore, it's important that you understand what the ability of the online software or application is and what it will and won't do for you. The existing websites are

• Resume.io

• Canva.com

• Jobseeker.com

• Zety.com

2.1.2 Drawbacks

- You still have to input your information onto the page or software for the resume builder to be able to complete your resume.
- Often you are restricted to use the builder's fonts and formats.
- Some builders are purely 100% online and don't have the functionality to allow the user to save their resume to other word processing software such as Microsoft Word, and often you can only print out what you have completed online. This makes it hard for individuals if they are applying for several different positions or if they want to adjust the resume for the different positions that they are applying for.
- Some of the resume builders only provide you with a PDF copy of your resume which again restricts you for further adaption or updating.

2.1.3 Implementation

There are several different ways that the existing systems can be implemented. Some of the most common approaches include:

1. Online resume builders: These are web-based tools that allow users to create a resume by entering their information into pre-designed templates. Users can typically choose from a variety of templates, and the tool generates a professional-looking resume based on the information entered.

2. Standalone software: Some resume builder systems are standalone software programs that users can download and install on their computers. These programs often have more advanced features and customization options compared to online resume builders.

3. Integration with job search websites: Some job search websites, such as LinkedIn and Indeed, offer resume builder tools as part of their platform. Users can create a resume directly within their account on the job search website and then use it to apply for jobs through the platform.

Regardless of the specific implementation, most resume builder systems use a combination of templates, input fields, and automated formatting to generate a professional-looking resume based on the information provided by the user.

CHAPTER 3
SYSTEM DESIGN

3. SYSTEM DESIGN

3.1 Proposed System Introduction

Creating a well-structured resume is a hectic task for students and many other job seekers. Therefore, we propose a web portal called MyFolio which is a resume builder which will ease the task of creating resumes. Resume plays a vital role in the employment of job seekers. This portal can be used to create resumes which are well structured. The user needs to select a resume template and they need to fill the personal details, educational details, work experiences, Skills details in the forms displayed in website. With all the user given input, the resume is generated which can be downloaded as pdf. These resumes are stored in the local storage of the browser hence it can be accessed later in the same device which is used while creating the resume. The user need not to login every time to create the resumes hence it will be the easier to create the resumes hassle free manner.

3.2 Proposed System Requirements

Software Requirements

- HTML
- CSS
- ReactJS
- Redux
- Js2PDF library
- **Tools:** Git, GitHub, Visual Studio Code

Hardware Requirements

- **Operating System:** Windows/Ubuntu/Linux
- **Hard Disk:** 10 GB
- **Ram:** 256MB
- **Input:** Keyboard & Mouse

3.3 Proposed System Architecture

The architecture of the proposed system describes about the working of the MyFolio Resume Builder. The user needs to select the Resume Template displayed to them on the website. After Selecting the Template, the user needs to fill all the necessary details like personal details, educational details, work experiences, skills which is displayed as step-wizard form. All the details given by the user will be stored in the Local Storage of the browser and the resume is generated which can be downloaded in the form of PDF.

1. **User interface:** The front-end of the resume builder is built using ReactJS, which is a JavaScript library for building user interfaces. It allows you to create reusable UI components and efficiently update the UI in response to user actions.

2. **State management:** To manage the state of the application, you can use Redux, which is a popular JavaScript library for managing the application state. It allows you to centralize the state of your application in a single store and manage it using reducers and actions.

3. **Local storage:** To store the user's resume data, you can use the local storage feature of the user's web browser. This allows you to store data on the user's device and retrieve it later, even if the user closes the browser or shuts down the computer.

4. **js2Pdf library:** To generate a PDF version of the resume, you can use a library like js2Pdf. This library allows you to create a PDF document from a HTML page and download it to the user's device.
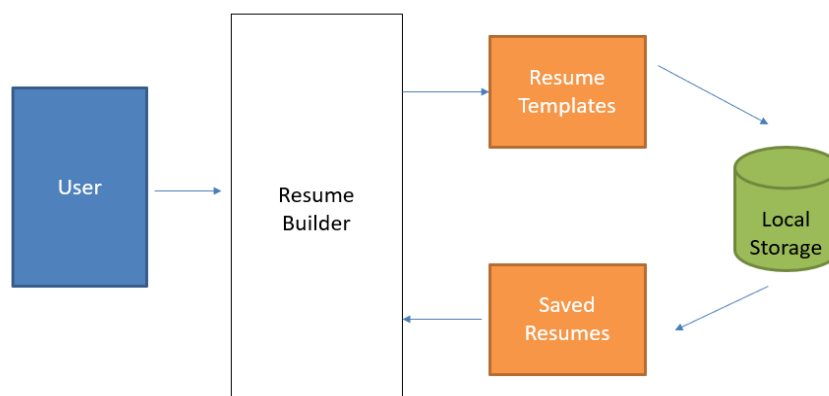

Fig 3.3.1 Architecture of Resume Builder

ReactJS

ReactJS is a popular JavaScript library for building user interfaces and is often used for building web applications. Here are some advantages of using ReactJS for a resume builder project:

1. Reusable components: One of the main benefits of React is the ability to create reusable components that can be easily shared and reused throughout the application. This can help to reduce code duplication and make it easier to maintain the codebase.

2. Fast rendering: React uses a virtual DOM (Document Object Model) to optimize the rendering of components, which can result in faster rendering times and a better user experience.

3. Easy to learn: React has a simple and intuitive syntax, making it relatively easy for developers to learn and use.

4. Strong community support: React has a large and active community of developers, which means there are many resources available for learning and troubleshooting.

5. Integration with other technologies: React can be easily integrated with other technologies and libraries, such as Redux for managing application state, and React Router for handling routing.

3.4 Advantages of Proposed Method

There are several advantages to using a resume builder project:

1. Time-saving: It can help you create a professional-looking resume quickly and easily, saving you time and effort.

2. Customization: With MyFolio, you can customize your resume to highlight your unique skills and experience, making it more likely to stand out to potential employers.

3. Professional design: It pre-designed templates and formatting options that can give your resume a polished, professional look.

4. Easy updates: If you need to make changes to your resume, a resume builder makes it easy to update and maintain.

5. Accessibility: With MyFolio, you can create and update your resume from any device with an internet connection, making it easy to access and edit from anywhere.

3.5 Implementation of Proposed Method

We have used react in our project which is One of the popular JavaScript libraries for developing front end web applications. Run the following commands in the Terminal to create and start a React app called "MyFolio":

```
npx create-react-app MyFolio
cd MyFolio
npm start
```
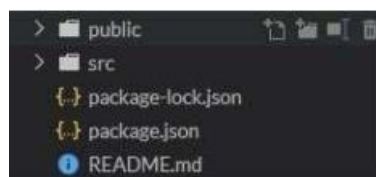
Basic Structure of a React Project

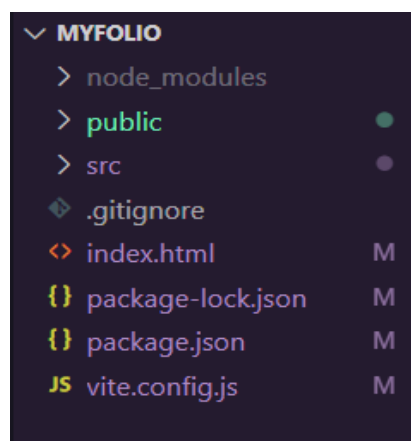Fig 3.5.1 Structure of React-App

The File Structure of MyFolio

Fig 3.5.2 File Structure of MyFolio

- The public folder contains files related to how the application will display on the client, the most important of those being index.html, which is the HTML template of our page.

- The src folder contains all of the JavaScript, CSS, and image files that will be compiled into a bundle file and injected into index.html.
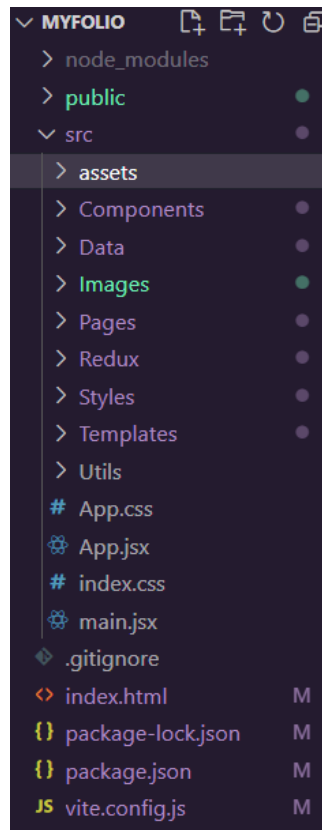
Fig 3.5.3 Contents of src folder

Source Code

Index.js

```
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App'
import './index.css'
import { store } from './Redux/store'



import { Provider} from 'react-redux';
```

```
ReactDOM.createRoot(document.getElementById('root')).render(
  <Provider store={store}>
     <App />
  </Provider>


);
```

App.js

```
import "./App.css";
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
import Home from "./Pages/Home";
import DetailsFilling from "./Pages/DetailsFilling";
import CheckSelectedId from "./Components/CheckSelectedId";
import AboutUs from "./Pages/AboutUs";
import MyResumes from "./Pages/MyResumes";



const App = () => {
  return (
    <Router>
     <Routes>

       {/* Home Page Route */}
       <Route exact path="/" element={<Home />} />


       {/* Details Filling Page */}
       <Route
        exact
        path="/template/fill-details"
        element={
          <CheckSelectedId>
           <DetailsFilling />
          </CheckSelectedId>
```

```
      }
    />


    {/* My Resumes Page */}
    <Route exact path="/my/resumes" element={<MyResumes />} />


    {/* About Us Page */}
    <Route exact path="/about-us" element={{<AboutUs />} />
  </Routes>
 </Router>
 );
};


export default App;
```

Home.jsx

```
import React from "react";
import Navbar from "../Components/Common/Navbar";
import "../Styles/Home.css";
import { templates } from "../Data/templates";
import BlackScreen from "../Components/BlackScreen";
import { Button, Stack } from "@mui/material";
import { connect } from "react-redux";
import { useNavigate } from "react-router-dom";
import { selectTemplate } from "../Redux/actions";
import logo from '../Images/resume-amico.svg';
const mapStateToProps = (state) => ({
 selectedTemplateId: state.selectedTemplateReducer.selectedTemplateId,
});
const mapDispatchToProps = (dispatch) => ({
 setSelectedTemplateId: (id) => dispatch(selectTemplate(id)),
});
```

```
const Home = (props) => {
  const navigate = useNavigate();

  const navigateToFillDetails = (id) => {
    props.setSelectedTemplateId(id);
    navigate("/template/fill-details");
  };
  return (
    <>
    <Navbar active={"Home"} />

      <div className="hero_section">
        <div className="hero_section-container">
          <div className="hero_section-container-left">
            <h1 className="home-title">MyFolio</h1>
            <p className="home-description">Create Your Resume By employing the best
practices and innovative tech, MyFolio boosts your chances of landing a better job.</p>
            <Button variant="contained" color="primary" href="#getStarted" >Get
Started</Button>
          </div>
          <div className="hero_section-container-right">
            <img src={logo} alt="Resume" />
          </div>
        </div>
      </div>
      <>
        <div className="home">
        <div className="home-templates-cont">
          <h2 id="getStarted" className="template-header-title">Choose a Template</h2>
          {/* <p className="template-select-text">Select one of the templates to get
started</p>
          <p className="template-select-text">There you go..</p> */}
          <Stack
            sx={{{
```

```
        width: "75%",
        display: "grid",
        gridTemplateColumns: {
          sm: "1fr 1fr",
          md: "1fr 1fr",
          lg: "1fr 1fr 0.3fr",
          xl: "1fr 1fr 1fr 1fr",
        },
        gridGap: "30px",
      }}>
      {templates.map((template) => {
       return (
        <div key={template.id} id="template" className="templates-img-cont">
         <img
          className="template-img"
          src={template.template_img}
          alt={template.template_name}
         />
         <BlackScreen />
         <Button
          className="use-template-btn"
          onClick={() => navigateToFillDetails(template.id)}
          size="medium"
          variant="contained">
          Use Template
         </Button>
        </div>
       );
      })}
     </Stack>
    </div>
   </div>
   </>
```

```
    </>
  );
};


export default connect(mapStateToProps, mapDispatchToProps)(Home);
```

DetailsFilling.jsx

```jsx
import React, { useState } from "react";
import Navbar from "../Components/Common/Navbar";
import DetailFillingSidebar from "../Components/DetailFillingSidebar";
import EducationComponent from "../Components/EducationComponent";
import KeySkillsComponent from "../Components/KeySkillsComponent";
import PersonalInfoComponent from "../Components/PersonalInfoComponent";
import PreviewComponent from "../Components/PreviewComponent";
import ProjectComponent from "../Components/ProjectComponent";
import WorkExperienceComponent from "../Components/WorkExperienceComponent";
import "../Styles/DetailsFilling.css";
const DetailsFilling = (props) => {
  const [tab, setTab] = useState(0);
  return (
    <div className="details-filling">
      <Navbar active={""} />
      {tab === 5 ? null : (
        <div className="details-filling-cont">
          <DetailFillingSidebar tab={tab} setTab={setTab} />
          {tab === 0 ? (
            <PersonalInfoComponent setTab={setTab} tab={tab} />
          ) : null}
          {tab === 4 ? <KeySkillsComponent setTab={setTab} tab={tab} /> : null}
          {tab === 1 ? (
            <WorkExperienceComponent setTab={setTab} tab={tab} />
          ) : null}
          {tab === 2 ? (
```

```
        <ProjectComponent setTab={setTab} tab={tab} />
      ) : null}
      {tab === 3 ? <EducationComponent setTab={setTab} tab={tab} /> : null}
    </div>
  )}



  {tab === 5 ? <PreviewComponent setTab={setTab} tab={tab} /> : null}
  </div>
 );
};


export default DetailsFilling;
```

MyResumes.jsx

```
import React, { useEffect, useState } from "react";
import Navbar from "../Components/Common/Navbar";
import "../Styles/MyResumes.css";
import { Button } from "@mui/material";
import BlackScreen from "../Components/BlackScreen";
import { templates } from "../Data/templates";
import SentimentVeryDissatisfiedIcon from "@mui/icons-
material/SentimentVeryDissatisfied";
import { Box } from "@mui/system";
import Grid from "@mui/material/Grid";
import { styled } from "@mui/material/styles";
import Paper from "@mui/material/Paper";
import JsPDF from "jspdf";
import { connect } from "react-redux";
import {
 addAllExperience,
 addEducation,
 addPersonalInfo,
 addAllProjects,
```

```
  editSkill,
  selectResume,
  selectTemplate,
} from "../Redux/actions";
import { useNavigate } from "react-router-dom";


const Item = styled(Paper)(({ theme }) => ({
  backgroundColor: theme.palette.mode === "dark" ? "#1A2027" : "#fff",
  ...theme.typography.body2,
  padding: theme.spacing(1),
  color: theme.palette.text.secondary,
}));


const mapStateToProps = (state) => ({
  selectedTemplateId: state.selectedTemplateReducer.selectedTemplateId,
});


const mapDispatchToProps = (dispatch) => ({
  setSelectedTemplateId: (id) => dispatch(selectTemplate(id)),
  setSelectedResumeId: (id) => dispatch(selectResume(id)),
  onAddPersonalInfo: (details) => dispatch(addPersonalInfo(details)),
  setAllExperience: (experiences) => dispatch(addAllExperience(experiences)),
  setAllProjects: (projects) => dispatch(addAllProjects(projects)),
  onAddEducation: (details) => dispatch(addEducation(details)),
  onEditSkill: (skills) => dispatch(editSkill(skills)),
});


const MyResumes = (props) => {
  const [resumes, setResumes] = useState([]);

  useEffect(() => {
    const newResumes = window.localStorage.getItem("resumes")
      ? JSON.parse(window.localStorage.getItem("resumes"))
      : [];
```

```
    setResumes(newResumes);
  }, []);


  const navigate = useNavigate();


  const getTemplate = (resume, index) => {
   let template = templates.find(
     (eachTemplate) => eachTemplate.id === resume.template_id
   );
   // console.log("resume",resume)
   // console.log("template", template);


    const TemplateComp = React.cloneElement(template.template, {
      personalinfo: resume.personalInfo,
      workexperience: resume.experiences,
      educationinfo: resume.educationInfo,
      projects: resume.projects,
      skills: resume.skills,
      key: resume.id,
      index: index,
    });


    return TemplateComp;
  };


  const deleteResume = (resume) => {
   let resumes = window.localStorage.getItem("resumes");


   let newResumes = JSON.parse(resumes);
   const newSetOfResumes = newResumes.filter((eachResume) => {
     return eachResume.id !== resume.id;
   });
```

```
    window.localStorage.setItem("resumes", JSON.stringify(newSetOfResumes));
    setResumes(newSetOfResumes);
  };


  const downloadResume = (id) => {
   // console.log(id);
   const report = new JsPDF("portrait", "pt", "a4");
   report.html(document.getElementById(`${id}report`)).then(() => {
    report.save(`resume.pdf`);
    // // console.log(resumes)
   });
  };


  const setUserData = (resume) => {
   // console.log(resume);
   //set personal info


   props.onAddPersonalInfo(resume.personalInfo);


   //set work experience


   props.setAllExperience(resume.experiences);


   //set project
   props.setAllProjects(resume.projects);


   //set education info


   props.onAddEducation(resume.educationInfo);


   //set skills


   props.onEditSkill(resume.skills);
  };
```

```
const navigateToFillDetails = (resume) => {
  props.setSelectedTemplateId(resume.template_id);
  props.setSelectedResumeId(resume.id);
  setUserData(resume);
  navigate("/template/fill-details");
};

// console.log(resumes);
return (
  <>
    <Navbar active={"My Resumes"} />
    <div className="my-resumes">
     <Box sx={{ flexGrow: 1 }}>
      <Grid
        container
        justifyContent="center"
        alignItems="center"
        className="grid"
      >
        {resumes.length >= 1 ? (
         resumes.map((resume, index) => {
           return (
            <Grid
              item
              className={`resume `}
              id={`${index}resume`}
              margin={2}
              key={index}
            >
              <Item id={`${index}ITEM`}>
                {getTemplate(resume, index)}
                <BlackScreen />
                <div className="use-template-btn-cont">
```

```
<Button
 // sx={{
 //  position: "relative",
 //  top: "-35vh",
 //  left: "0px",
 //  transform: "inherit",
 // }}
 className="use-template-btn"
 onClick={() => {
   downloadResume(index);
 }}
 size="medium"
 variant="contained"
>
 Download
</Button>
<Button
 className="use-template-btn"
 onClick={() => {
   deleteResume(resume);
 }}
 size="medium"
 variant="contained"
>
 Delete
</Button>
<Button
 className="use-template-btn"
 onClick={() => navigateToFillDetails(resume)}
 size="medium"
 variant="contained"
>
 Edit Template
</Button>
```

```
        </div>
      </Item>
    </Grid>
  );
})
) : (
  <div className="no-resumes-container">
   <SentimentVeryDissatisfiedIcon fontSize="large" />
   <p className="no-resumes-text">
     You do not have any Resumes yet. Make One to view it here.
   </p>
  </div>
 )}
 </Grid>
</Box>
</div>
</>
);
};

export default connect(mapStateToProps, mapDispatchToProps)(MyResumes);
```

Template.jsx

```
import { Container, Paper } from "@mui/material";
import React from "react";
import "../Styles/Template1.css";
import TemplateHeader from "../Components/TemplateHeader";
import TemplateHeading from "../Components/TemplateHeading";
import TemplateOneExperienceComponent from
"../Components/TemplateOneExperienceComponent";
import { data } from "../Data/data";
import TemplateEducationComponent from "../Components/TemplateEducationComponent";
import TemplateKeySkillComponent from "../Components/TemplateKeySkillComponent";
import TemplateOneProjectComponent from
```

```
"../Components/TemplateOneProjectComponent";

const Template1 = (props) => {
  console.log(
   props.personalinfo,
   props.workexperience,
   props.projects,
   props.educationinfo,
   props.skills
  );
  const personalinfo = props.personalinfo
   ? props.personalinfo
   : data.personal_info;
  const workexperience = props.workexperience
   ? props.workexperience
   : data.work_experience;
  const proj = props.projects ? props.projects : data.projects;
  const educationinfo = props.educationinfo
   ? props.educationinfo
   : data.education_details;
  const skills = props.skills ? props.skills : data.key_skills;

  console.log(props.index);
  return (
   <Paper
    style={{
     padding: "0 0 20px 0",
    }}
    sx={{
     width: {
       xs: "400px",
       sm: "450px",
       md: "500px",
       lg: "550px",
```

```
      xl: "600px",
     },
    height: {
     xs: "500px",
     sm: "550px",
     md: "600px",
     lg: "650px",
     xl: "700px",
    },
   }}
   id={`${props.index}report`}
   elevation={2}>
   <TemplateHeader
    primaryColor={"tomato"}
    secondaryColor={"black"}
    bgColor={"white"}
    personalInfo={personalinfo}
    workExperience={workexperience}
   />
   <Container>
    <TemplateHeading color={"tomato"} title={"Experience"} />
    <ul style={{ paddingBottom: 10 }}>
     {workexperience.map((experience, index) => {
      return (
       <TemplateOneExperienceComponent
        key={index}
        experience={experience}
       />
      );
     })}
    </ul>
    <TemplateHeading color={"tomato"} title={"Projects"} />
    <ul style={{ paddingBottom: 10 }}>
     {proj.map((project, index) => {
```

```
      return (
        <TemplateOneProjectComponent
          key={index}
          project={project}
        />
      );
    })}
  </ul>
  <TemplateHeading color={"tomato"} title={"Education"} />
  <TemplateEducationComponent education={educationinfo} />
  <TemplateHeading color={"tomato"} title={"Skills"} />
  <ul style={{ marginBottom: 10 }}>
    {skills.map((skill, index) => {
      return <TemplateKeySkillComponent key={index} skill={skill} />;
    })}
  </ul>
  </Container>
  </Paper>
 );
};

export default Template1;
```

CHAPTER 4
RESULTS AND DISCUSSION

4. RESULTS AND DISCUSSION

4.1 Performance Analysis

There are several factors that can impact the performance of our resume builder website:

1. Page load time: The faster the page loads, the better the user experience will be. A slow-loading website can be frustrating for users and may lead to them leaving the site.
2. Usability: MyFolio is easy to navigate and use, with clear instructions and a simple layout.
3. Functionality: MyFolio has all the necessary features and tools that users need to create a professional-looking resume. This includes templates, formatting options, and the ability to save and download the completed resume.
4. Compatibility: The website is compatible with a variety of devices and browsers, so users can access it from any device.

4.2 Result and Screenshots

The Home Page of the MyFolio Resume Builder
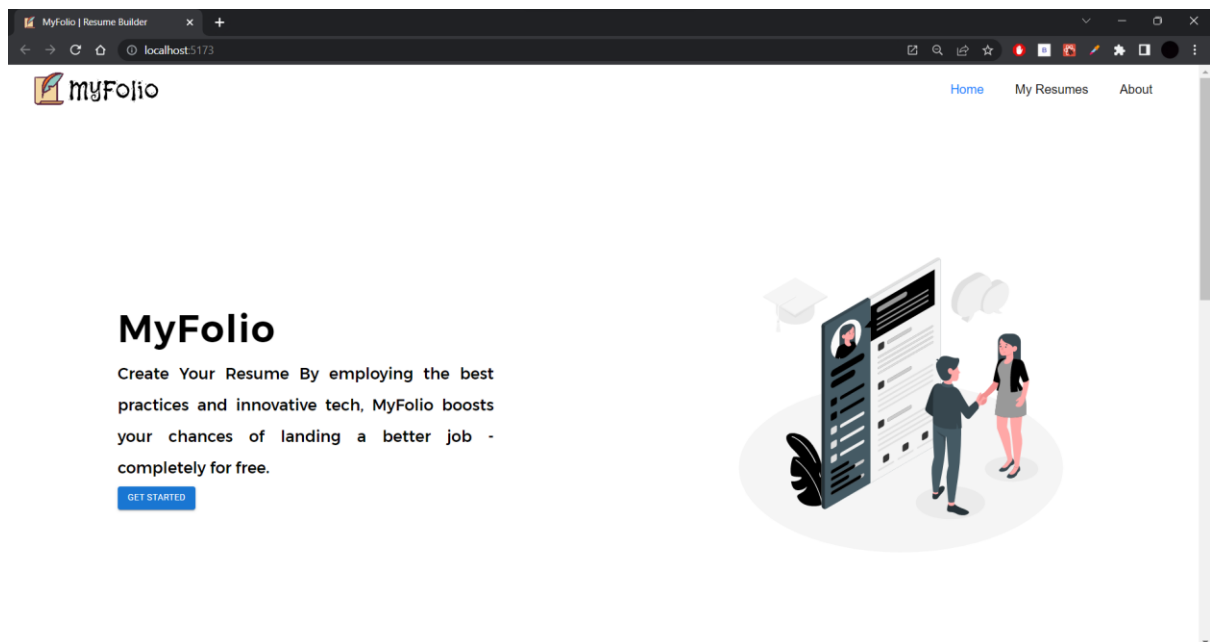
Fig 4.2.1 Home Page of MyFolio

The Templates are displayed to the user on the website
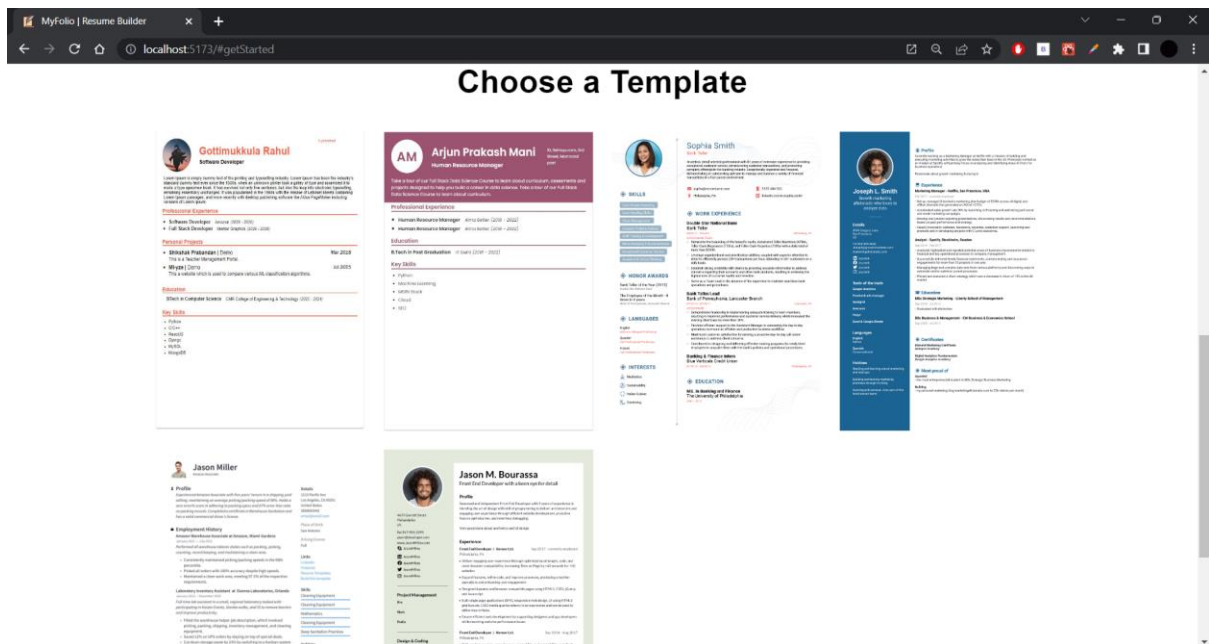
Fig 4.2.2 Resume Templates

The user needs to fill the personal information in the form

Fig 4.2.3 Personal Info Form

The user needs to fill the work experience details

Fig 4.2.4 Work Experience Form

The user needs to fill the project details

Fig 4.2.5 Project Details Form

The user needs to fill the educational details

Fig 4.2.6 Education Details Form

The user needs to skills details

Fig 4.2.7 Key Skills Form

The resume is generated with the given input and can be saved as pdf after giving it a name.
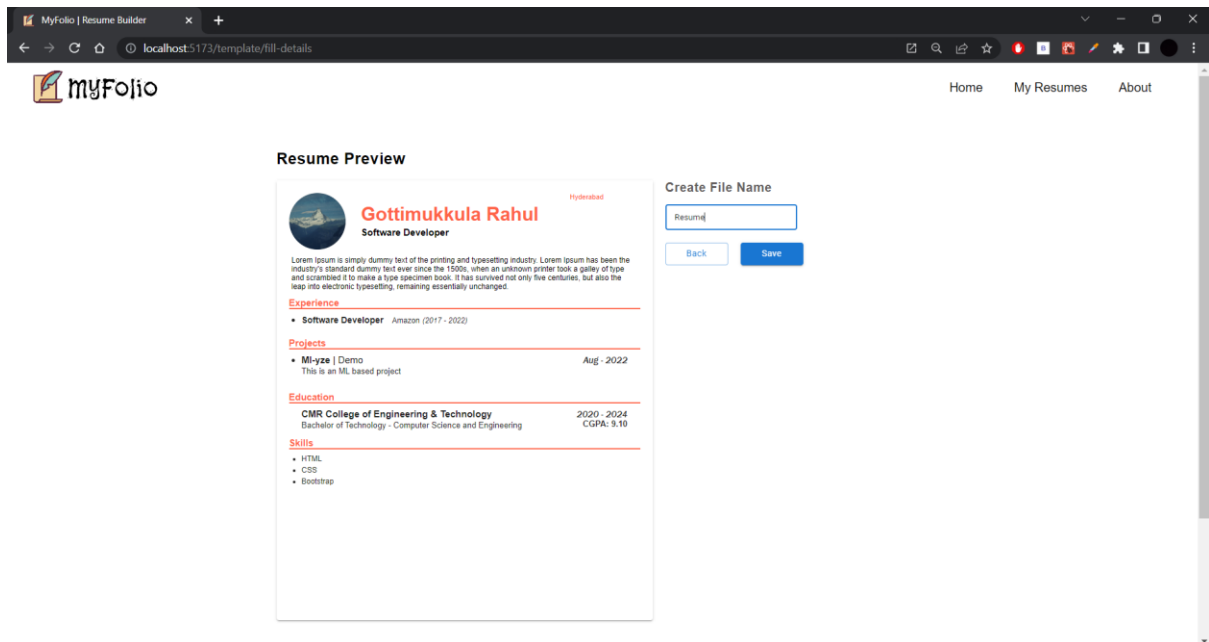

Fig 4.2.8 Preview of Resume

The resumes created are saved in the local storage which can be used to access the previous created resumes by the user.
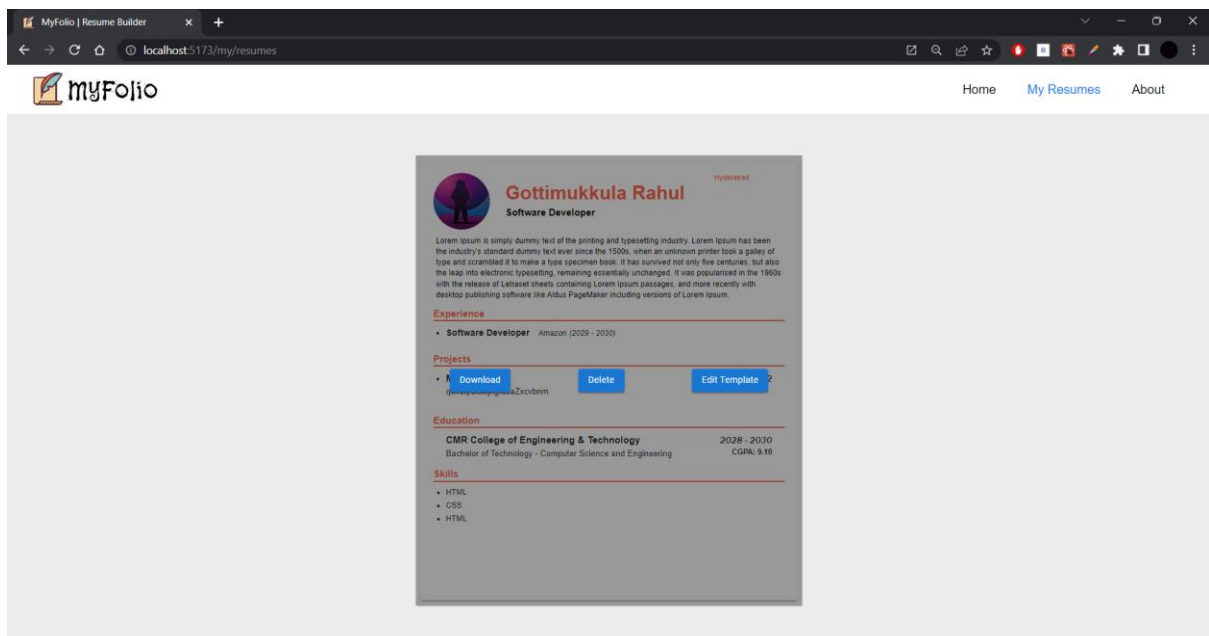

Fig 4.2.9 Saved Resumes

CHAPTER 5
CONCLUSION

5. CONCLUSION

5.1 Conclusion

In conclusion, MyFolio - resume builder is a valuable tool for job seekers as it can help them create professional-looking and effective resumes more easily. It can save users time and effort by providing templates, customization options, and the ability to import and export data, rather than having to create a resume from scratch. It can also help users stand out from the competition by offering unique and visually appealing templates, as well as customization options to tailor their resume to specific job openings. It can be a useful resource for career counselors, recruiters, and other professionals who work with job seekers, as it can help them create strong and effective resumes for their clients or candidates. This project is easy to access and does not require login details to download the resume. unlike other websites where login is required to download the resume. Since the language used in our project is react which is simple, easy to understand reusable components and is an open-source platform.

CHAPTER 6
REFERENCES

6. REFERENCES

[1] Anindhynaojha, Upasanapatnaik and relhasanskar, Data analytics on placement data in south asian University (2 August 2017) IEEE publication.

[2] Sommathmazumdar and danielseybold, A survey on data storage and placement methodologies (22 January 2019) Springer Nature publication.

[3] Anderson, N. H., & Barrios, A. A. Primacy effects in personality impression formation. Journal of Abnormal and Social Psychology, 63, 346-350.

[4] Field, H. S., & Holley, W. H. Resume preparation: An empirical study of personnel managers' perceptions. Vocational Guidance Quarterly, 24, 229-237.

[5] Holley, W. H., Higgins, E., & Speights, S. Resumes and cover letters: What do HR managers really want? Personnel, 65, 49-51.

[6] Knouse, S. B., Giacalone, R. A., & Pollard, H. Impression management in the resume and its cover letter. Journal of Business and Psychology, 3, 242-249.

[7] Penrose, J. M. A discrepancy analysis of the job-getting process and a study of resume techniques. The Journal of Business Communication, 21, 5-15.

[8] Toth, C. Effect of resume format on applicant selection for job interviews. Applied H.R.M. Research, 4, 115-125. Waung, M., Hymes, R. W., & Beatty, J. E. (2014).

[9] "The Best Resume Templates for 2021" by The Balance Careers: This article features a selection of professional and visually appealing resume templates that can be used as a starting point for building a resume.

[10] "The Best Resume Builders of 2021" by PCMag: This article reviews and compares a selection of resume builder tools, including their features, pricing, and ease of use.