NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

A MINI PROJECT REPORT

on

WEB DESIGN TECHNOLOGIES (24CSE361)

TITLE:JOB LISTING WEBSITE

Submitted by

NAME:A AMRUTHA
USN:1NH24CS001

Under the guidance of
GUIDE NAME:Ms. M. THANGA SUBHA DEVI
Designation:Sr.Assistant professor

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Academic Year: 2025-26 (ODD SEM)

NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

CERTIFICATE

This is to certify that the mini project work titled "**JOB LISTING WEBSITE"** is a bonafide work carried out by **A AMRUTHA (1NH24CS001)** in partial fulfillment of the degree of **Bachelor of Engineering** in **Computer Science and Engineering** of the New Horizon College of Engineering during the year **2025-2026.**

Signature of Guide

Signature of HOD

SEMESTER END EXAMINATION

Name of the Examiner *Signature with date*

1._____ _____

2._____ _____

ABSTRACT

This job-listing platform is a scalable, data-driven web application engineered to optimize recruitment workflows through automated matching and efficient information retrieval. The system integrates a secure user authentication framework, role-based access controls, and a modular architecture supporting employer, job-seeker, and admin interfaces. Job postings, user profiles, and application data are managed through a normalized relational database with optimized indexing for high-performance querying.

Advanced search functionality—powered by full-text indexing, filtering algorithms, and relevance ranking—enables rapid discovery of opportunities across industries. The platform employs asynchronous processing for notification delivery, application updates, and background analytics tasks. RESTful APIs (or GraphQL endpoints) provide interoperability with third-party HR systems and external job-distribution networks.

Built with responsive UI/UX components and secure communication protocols, the system ensures reliable data integrity, scalability, and seamless cross-device accessibility, delivering a robust and efficient digital hiring ecosystem.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, who's constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing gratitude to **Dr. Mohan Manghnani**, Chairman, New Horizon Educational Institutions, for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to **Dr. Manjunatha,** Principal, New Horizon College of Engineering, for the constant support and encouragement.

I would like to thank **Dr. Anandhi R J**, Professor and Dean-Academics, NHCE, for her valuable guidance.

I would also like to thank **Dr. B. Rajalakshmi,** Professor and HOD, Department of Computer Science and Engineering, for the constant support.

I also express my gratitude to **Ms.M.THANGA SUBHA DEVI** , DesignationSr.Assistant professor, Department of Computer Science and Engineering, my mini project reviewer, for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

Student
Name :A AMRUTHA
USN:1NH24CS 001

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION

The CareerConnect website is a frontend job-listing platform built using HTML, CSS, and JavaScript. It allows users to browse and filter jobs through an interactive search bar and category selector, with job data dynamically rendered from a JavaScript array. The layout features a responsive design, clean UI components, and interactive job cards that enhance usability. Although fully client-side, the system is structured to support future integration with backend services or APIs.

Many job seekers struggle to quickly find relevant job opportunities due to scattered listings, poor filtering options, and cluttered interfaces on existing platforms. Similarly, employers often face challenges in presenting job information in a clear and accessible way. The lack of a simple, responsive, and efficient system makes job search time-consuming and inconvenient.

The CareerConnect project aims to solve this problem by providing a lightweight, user-friendly frontend platform where job seekers can easily browse and filter job postings. Through dynamic search functionality and category-based filtering, the system simplifies job discovery and presents information in an organized, responsive layout. This solution creates a clean and accessible interface that improves the job-searching experience while offering a foundation for future expansion into a full-scale recruitment system.

1.2 OBJECTIVES

1. User-Friendly Interface

To develop a clean, intuitive, and responsive interface that allows users to easily navigate and browse job opportunities across various devices.

2. Dynamic Job Rendering

To use JavaScript for dynamically generating job cards from structured data, enabling real-time updates without reloading the page.

3. Efficient Search and Filtering

To implement keyword search and category-based filtering so users can quickly locate relevant job postings based on their preferences.

4. Modern UI Design

To apply modern HTML and CSS styling techniques to create visually appealing layouts, interactive elements, and improved user experience.

5. Structured Job Data Management

To maintain job details (title, company, location, category, and tags) in an organized format that supports easy modification and scalability.

6. Extendable Frontend Framework

To build the platform in a way that can later integrate with backend systems, APIs, databases, or authentication modules for full deployment.

1.3 METHODOLOGIES TO BE FOLLOWED

1. Requirement Analysis

Identify the key functionalities needed for a job-listing platform, including job search, filtering, and display.

Define user requirements for both job seekers and employers.

Outline the data structure for job postings (title, company, location, category, tags, link).

2. System Design

Design the overall layout and UI components, including the navigation bar, filter bar, job cards, and footer.

Create a responsive design to ensure compatibility across devices and screen sizes.

Plan the data flow between the user interface and JavaScript-based dynamic rendering.

3. Frontend Development

Implement the HTML structure for the navigation bar, filter bar, job listings, and footer.

Apply CSS for styling, responsive grids, hover effects, and UI aesthetics.

Use JavaScript to dynamically render job cards, implement search, and category-based filtering.

4. Testing and Debugging

Test functionality of the search and filter features with different inputs.

Check responsiveness across various devices and browsers.

Identify and fix bugs in HTML, CSS, or JavaScript code to ensure smooth operation.

5. Documentation

Prepare documentation including introduction, problem definition, objectives, methodology, and code explanations.

CHAPTER 2

FUNDAMENTALS OF THE LANGUAGES USED

2.1 HTML

->HTML is the standard markup language used to structure content on the web.

->It defines elements such as headings, paragraphs, links, forms, and containers.

->In CareerConnect, HTML is used to create the navigation bar, filter bar, job listing section, job cards, and footer.

->Key concepts used include semantic tags (<nav>, <section>, <footer>), attributes (id, class, placeholder), and links (<a>).

2.2 HTML TAGS

1. Document Structure Tags

<html>
<head>
<title>
<meta>
<style>
<body>

2. Layout & Semantic Tags

<nav> – navigation bar
<section> – job listings section
<footer> – footer section

3. Text & Heading Tags

<h1> – website title
<h3> – job title
<p> – company name, location, messages

4. List Tags

 – navigation menu
 – menu items

5. Form & Input Tags

<input> – search bar

<select> – category dropdown

<option> – dropdown options

<button> – search button

6. Container & Inline Tags

 – layout containers (filter bar, job card, tags)

 – job skill tags

7. Link & Script Tags

<a> – navigation links & apply button

<script> – JavaScript code

2.3 CSS

CSS is used to control the presentation and layout of HTML elements.

It allows customization of colors, fonts, spacing, borders, and responsive layouts.

In CareerConnect, CSS is used for styling the navigation bar, job cards, filter bar, hover effects, and responsive grids.

Key concepts include selectors, properties, box model, flexbox, grid layout, transitions, and media queries.

2.4 JAVASCRIPT

JavaScript is a programming language used to add interactivity and dynamic behavior to web pages.

In CareerConnect, JS handles dynamic rendering of job cards, real-time search, and category-based filtering.

Key concepts used include arrays of objects, DOM manipulation (document.createElement, appendChild), event handling (onclick), conditional statements, loops, and functions.

JavaScript allows the application to update content without refreshing the page, improving user experience.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 HARDWARE REQUIREMENTS

1. Development Environment

Processor: Intel Core i3 or equivalent

RAM: Minimum 4 GB

Storage: At least 100 MB free space for project files and code editor

Display: 1280×720 resolution or higher

Peripherals: Keyboard and mouse

2. Testing and Usage Environment

Processor: Dual-core CPU or higher

RAM: 2 GB or more

Storage: Minimal (browser cache and temporary files)

Display: Supports standard web resolutions (desktop or mobile screens)

Web Browser: Latest version of Chrome, Firefox, Edge, or Safari

3. Optional (for Advanced Development)

SSD for faster file access and loading

Multi-monitor setup for better code and design workflow

Development tools such as VS Code, browser developer tools, and version control (Git)

3.1 SOFTWARE REQUIREMENTS

The CareerConnect platform is a frontend web application that relies on standard web technologies. The software requirements ensure smooth development, testing, and deployment.

1. Operating System

Windows 10 or later / macOS / Linux

Any OS capable of running modern web browsers and code editors

2. Web Browser

Latest versions of Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari

Must support HTML5, CSS3, and JavaScript ES6 standards

3. Development Tools

Code Editor: Visual Studio Code, Sublime Text, Atom, or any text editor

Browser Developer Tools: Built-in DevTools for debugging and inspecting HTML, CSS, and JS

4. Supporting Software (Optional)

Version Control System: Git and GitHub for project management and collaboration

Image editing tools (e.g., Photoshop, Figma) for UI/UX design

Local server (e.g., XAMPP, WAMP, or live-server VS Code extension) if needed for testing dynamic content or future backend integration

CHAPTER 4

DESIGN

4.1 DESIGN GOALS

The design of the CareerConnect job-listing website is guided by the following key objectives:

1. User-Friendly Interface

Create an intuitive and easy-to-navigate layout.

Ensure that users can quickly find, filter, and apply for jobs without confusion.

2. Responsive Design

Make the website fully functional across various devices and screen sizes (desktops, tablets, smartphones).

Use flexible layouts and CSS media queries for optimal viewing experiences.

3. Dynamic Content Rendering

Implement real-time job listings using JavaScript to allow updates without page reloads.

Ensure search and filter functionality provides instant results.

4. Visual Appeal

Use modern UI design principles with consistent colors, fonts, spacing, and interactive elements.

Include hover effects, tags, and buttons that improve engagement and readability.

5. Scalability

Design the code and data structures in a modular way so future enhancements (like backend integration or user authentication) can be easily added.

6. Performance Optimization

Keep the website lightweight and fast-loading.

Optimize DOM manipulation and minimize unnecessary scripts or heavy resources.

Appendix A

```
<html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>CareerConnect - Job Listings</title>
  <style>
    /* ===== GENERAL STYLES ===== */
    * {
      box-sizing: border-box;
    }
    body {
      font-family: 'Poppins', sans-serif;
      background-color: #f5f7fa;
      margin: 0;
      padding: 0;
    }

    /* ===== NAVIGATION BAR ===== */
    nav {
      background-color: #2d3e50;
      color: white;
      display: flex;
      justify-content: space-between;
      align-items: center;
      padding: 1rem 2rem;
    }
    nav h1 {
      margin: 0;
      font-size: 1.5rem;
```

```
nav ul {
    list-style: none;
    display: flex;
    margin: 0;
    padding: 0;
  }
  nav li {
    margin-left: 1.5rem;
  }
  nav a {
    color: white;
    text-decoration: none;
    transition: color 0.3s;
  }
  nav a:hover {
    color: #ffcc00;
  }

  /* ===== FILTER BAR ===== */
  .filter-bar {
    display: flex;
    justify-content: center;
    align-items: center;
    background-color: white;
    padding: 1rem;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
    margin: 1.5rem auto;
    width: 90%;
    border-radius: 6px;
    flex-wrap: wrap;
```

```
gap: 10px;
  }

  .filter-bar input,
  .filter-bar select {
    padding: 10px;
    border-radius: 5px;
    border: 1px solid #ccc;
    font-size: 1rem;
  }

  .filter-bar button {
    background-color: #2d3e50;
    color: white;
    border: none;
    padding: 10px 15px;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s;
  }

  .filter-bar button:hover {
    background-color: #1e2a38;
  }

  /* ===== JOB LISTINGS ===== */
  .job-listings {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
    gap: 1.5rem;
    padding: 0 2rem;
```

```
}

.job-card {
  background: white;
  padding: 1.5rem;
  border-radius: 8px;
  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
  transition: transform 0.2s ease, box-shadow 0.2s ease;
}

.job-card:hover {
  transform: translateY(-5px);
  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.15);
}

.job-card h3 {
  margin-top: 0;
  color: #2d3e50;
}

.job-card .company {
  color: #666;
  font-size: 0.95rem;
  margin-bottom: 0.5rem;
}

.job-card .tags {
  margin-top: 10px;
}
```

```
.tag {
    display: inline-block;
    background-color: #e9ecef;
    color: #333;
    padding: 5px 10px;
    margin-right: 5px;
    border-radius: 20px;
    font-size: 0.8rem;
  }

  .apply-btn {
    margin-top: 1rem;
    display: inline-block;
    background-color: #28a745;
    color: white;
    text-decoration: none;
    padding: 8px 15px;
    border-radius: 5px;
    transition: background 0.2s;
  }

  .apply-btn:hover {
    background-color: #1f883d;
  }

  /* ===== FOOTER ===== */
  footer {
    text-align: center;
    background-color: #2d3e50;
    color: white;
    padding: 1rem;
```

```
margin-top: 2rem;
   }
  </style>
</head>
<body>

  <!-- ===== NAVIGATION ===== -->
  <nav>
    <h1>CareerConnect</h1>
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">Jobs</a></li>
      <li><a href="#">Post a Job</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </nav>

  <!-- ===== FILTER BAR ===== -->
  <div class="filter-bar">
    <input type="text" id="searchInput" placeholder="Search jobs..." />
    <select id="categorySelect">
      <option value="">All Categories</option>
      <option value="Tech">Tech</option>
      <option value="Design">Design</option>
      <option value="Management">Management</option>
      <option value="Marketing">Marketing</option>
    </select>
    <button onclick="filterJobs()">Search</button>
  </div>
```

```
<!-- ===== JOB LISTINGS ===== -->
  <section class="job-listings" id="jobContainer"></section>

  <footer>
    &copy; 2025 CareerConnect. All Rights Reserved.
  </footer>

  <script>
    const jobs = [
      { title: "Frontend Developer", company: "TechNova", category: "Tech", location: "Remote", tags: ["JavaScript", "React"], link: "#" },
      { title: "Graphic Designer", company: "Designify", category: "Design", location: "New York, USA", tags: ["Figma", "Photoshop"], link: "#" },
      { title: "Product Manager", company: "InnovateHub", category: "Management", location: "London, UK", tags: ["Leadership", "Agile"], link: "#" },
      { title: "Digital Marketer", company: "BrandBoost", category: "Marketing", location: "Toronto, Canada", tags: ["SEO", "Content"], link: "#" },
      { title: "Backend Engineer", company: "CloudSync", category: "Tech", location: "Berlin, Germany", tags: ["Node.js", "AWS"], link: "#" },
    ];

    const container = document.getElementById("jobContainer");

    function displayJobs(list) {
      container.innerHTML = "";
      if (list.length === 0) {
        container.innerHTML = "<p>No matching jobs found.</p>";
        return;
      }
      list.forEach(job => {
        const card = document.createElement("div");
```

```
card.classList.add("job-card");
        card.innerHTML = `
          <h3>${job.title}</h3>
          <p class="company">${job.company} - ${job.location}</p>
          <div class="tags">
            ${job.tags.map(tag =>      <span class="tag">${tag}</span>     ).join("")}
          </div>
          <a href="${job.link}" class="apply-btn">Apply Now</a>
        `;
        container.appendChild(card);
      });
    }

    function filterJobs() {
      const searchValue =
document.getElementById("searchInput").value.toLowerCase();
      const categoryValue = document.getElementById("categorySelect").value;
      const filtered = jobs.filter(job =>
        (job.title.toLowerCase().includes(searchValue) ||
         job.company.toLowerCase().includes(searchValue) ||
         job.location.toLowerCase().includes(searchValue)) &&
        (categoryValue === "" || job.category === categoryValue)
      );
      displayJobs(filtered);
    }

    // Display all jobs initially
    displayJobs(jobs);
  </script>
</body>
</html>
```

CODE SNIPPET

```
function filterJobs() {
  const searchValue = document.getElementById("searchInput").value.toLowerCase();
  const categoryValue = document.getElementById("categorySelect").value;
  const filtered = jobs.filter(job =>
    (job.title.toLowerCase().includes(searchValue) ||
     job.company.toLowerCase().includes(searchValue) ||
     job.location.toLowerCase().includes(searchValue)) &&
    (categoryValue === "" || job.category === categoryValue)
  );
  displayJobs(filtered);
}
```

CHAPTER 5

IMPLEMENTATION

The CareerConnect job-listing website is implemented using HTML, CSS, and JavaScript as a client-side web application. The implementation covers the following aspects:

1. Frontend Structure (HTML)

Navigation Bar: Provides links to Home, Jobs, Post a Job, and Contact pages.

Filter Bar: Includes a search input and category dropdown to allow job filtering.

Job Listings Section: Dynamically populated job cards displaying job title, company, location, tags, and an apply button.

Footer: Contains copyright and additional information.

2. Styling and Layout (CSS)

Responsive Grid: Job listings are arranged in a grid layout that adapts to different screen sizes.

Card Design: Each job card is styled with shadows, rounded corners, and hover effects to enhance interactivity.

Filter Bar Styling: Input fields, dropdowns, and buttons are styled for consistency and usability.

Color Scheme & Typography: A professional palette and readable fonts improve the user experience.

3. Dynamic Functionality (JavaScript)

Job Data Management: Jobs are stored in a JavaScript array of objects with fields for title, company, location, category, tags, and link.

Dynamic Rendering: The displayJobs() function creates job cards dynamically and appends them to the DOM.

Filtering Mechanism: The filterJobs() function filters jobs based on search keywords and category selection.

Interactive Elements: Apply buttons, hover effects, and instant updates improve usability.

4. Client-Side Processing

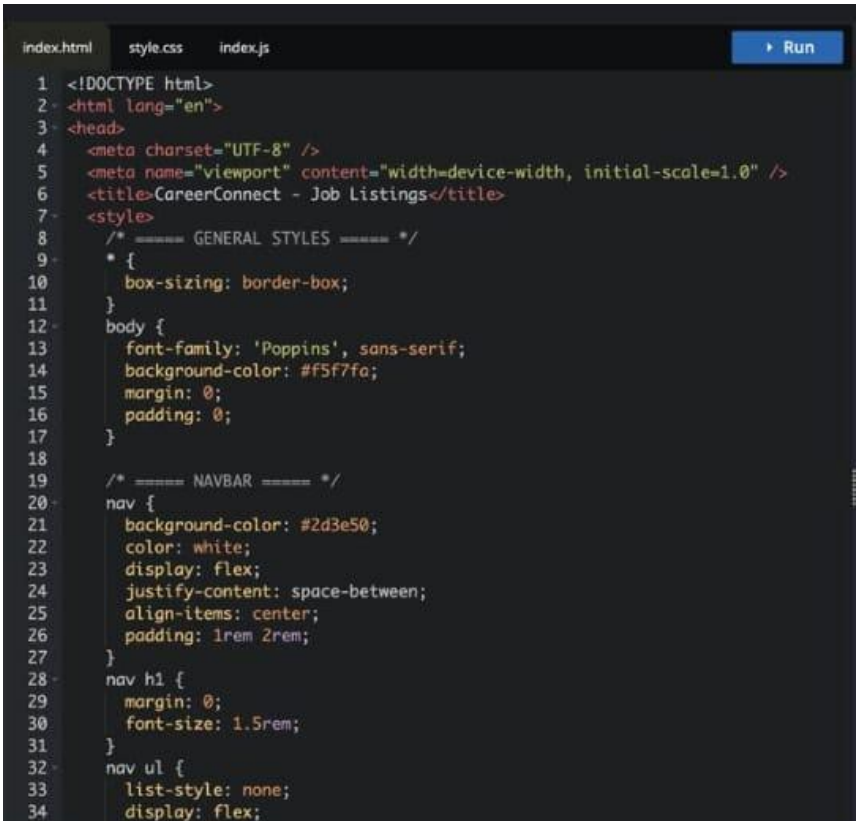
All filtering and rendering are handled on the client side without requiring a backend.

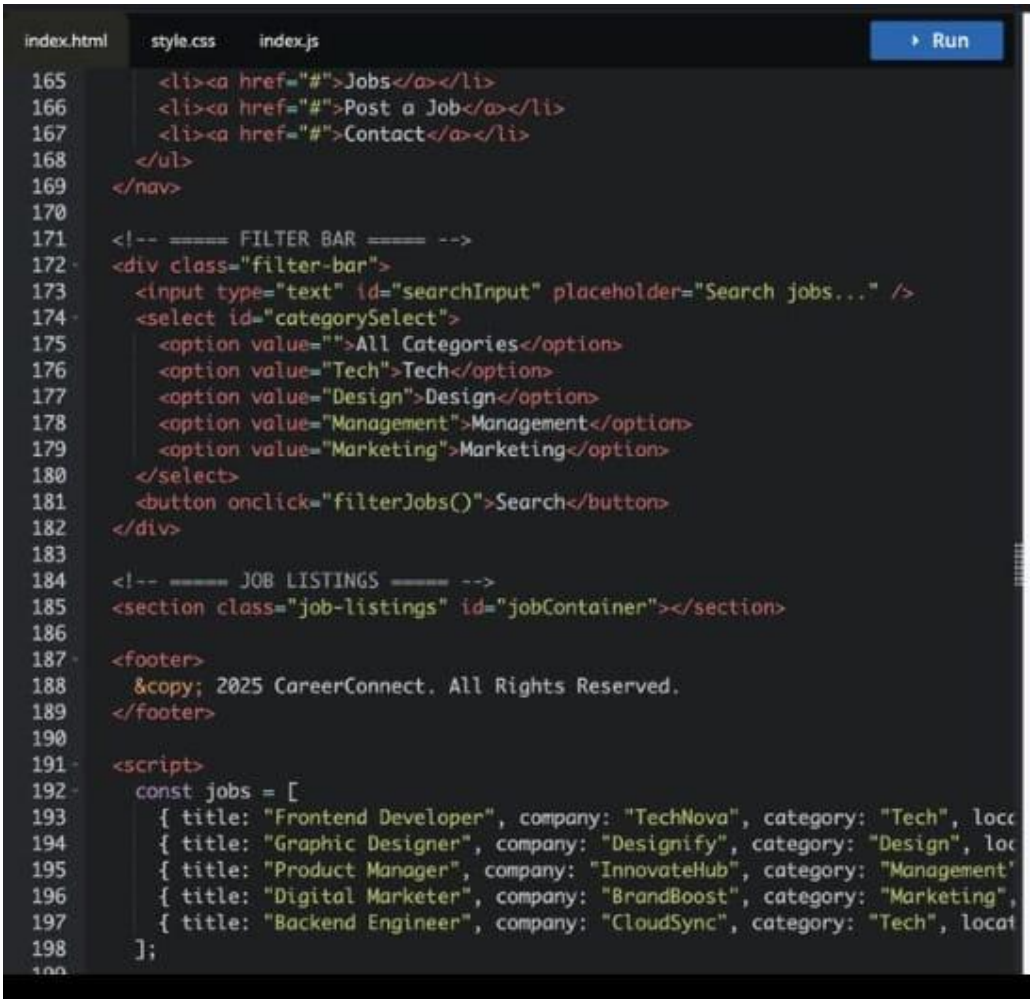
Provides fast, real-time updates as users type in search queries or select categories.

5. Scalability and Future Enhancements

The code is modular and structured to allow integration with backend databases or APIs for real-time job posting, user authentication, and employer dashboards.

Additional features like pagination, advanced search filters, and notifications can be added easily.


index.html
style.css
index.js
Run
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>CareerConnect - Job Listings</title>
<style>
/* ===== GENERAL STYLES ===== */
* {
box-sizing: border-box;
}
body {
font-family: 'Poppins', sans-serif;
background-color: #f5f7fa;
margin: 0;
padding: 0;
}
/* ===== NAVBAR ===== */
nav {
background-color: #2d3e50;
color: white;
display: flex;
justify-content: space-between;
align-items: center;
padding: 1rem 2rem;
}
nav h1 {
margin: 0;
font-size: 1.5rem;
}
nav ul {
list-style: none;
display: flex;


index.html
style.css
index.js
Run
Jobs
Post a Job
Contact

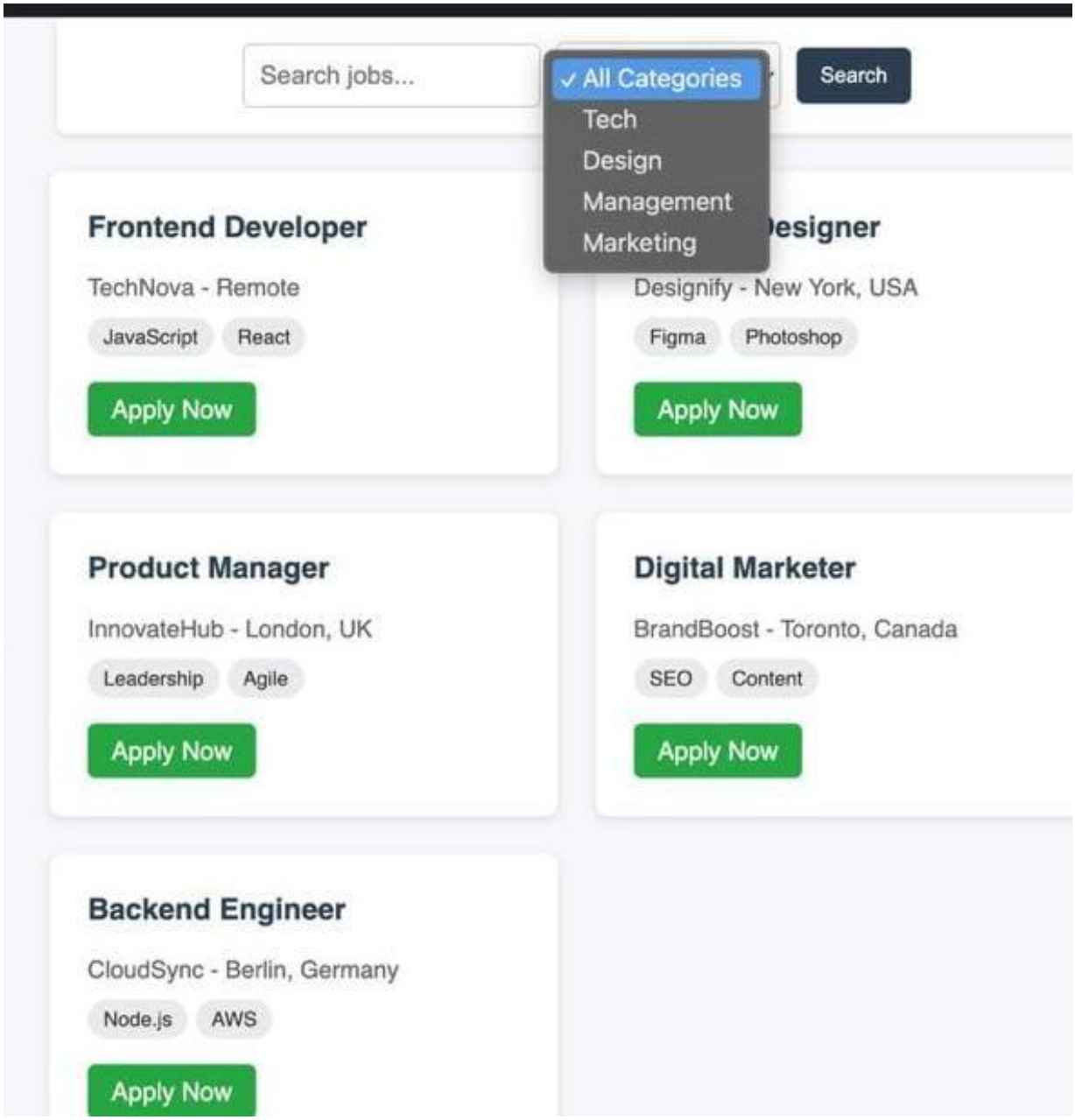
</nav>
<!-- ===== FILTER BAR ===== -->

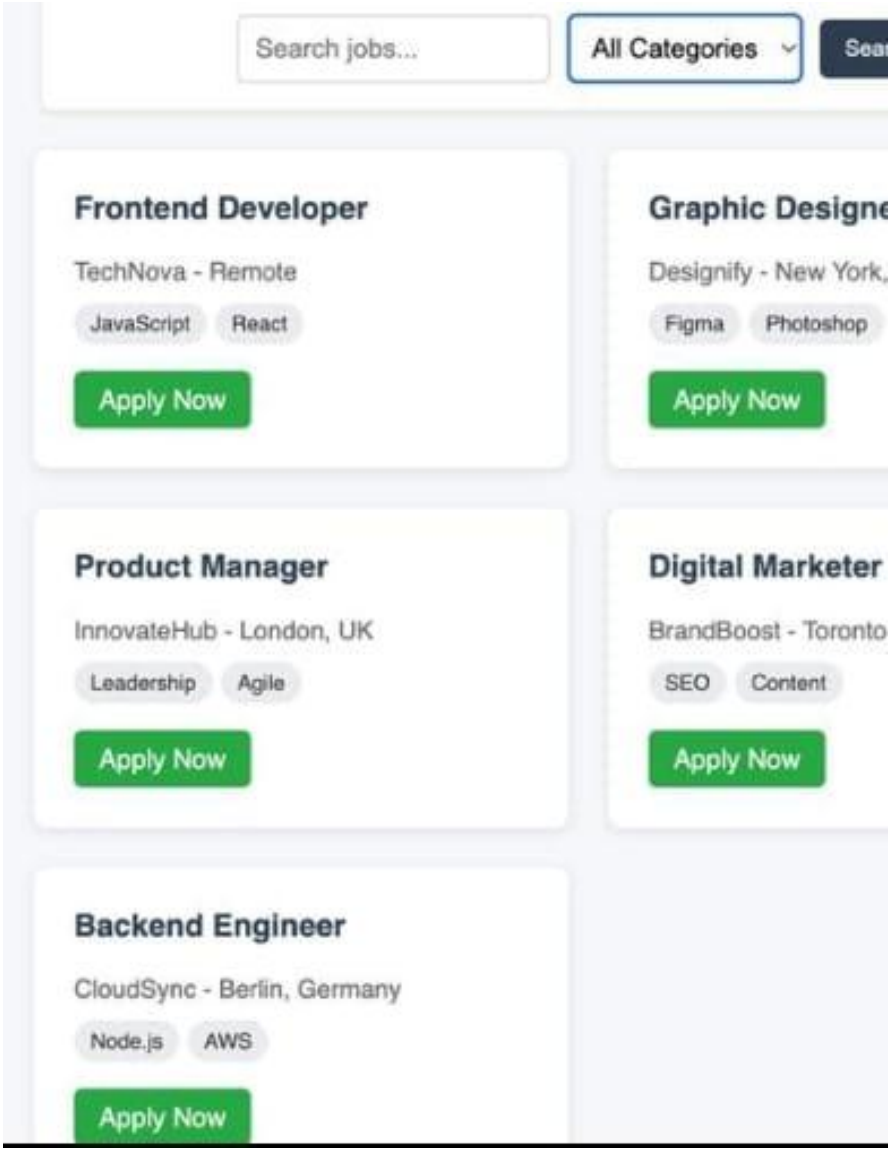
<input type="text" id="searchInput" placeholder="Search jobs..." />
<select id="categorySelect">
<option value="">All Categories</option>
<option value="Tech">Tech</option>
<option value="Design">Design</option>
<option value="Management">Management</option>
<option value="Marketing">Marketing</option>
</select>
<button onclick="filterJobs()">Search</button>

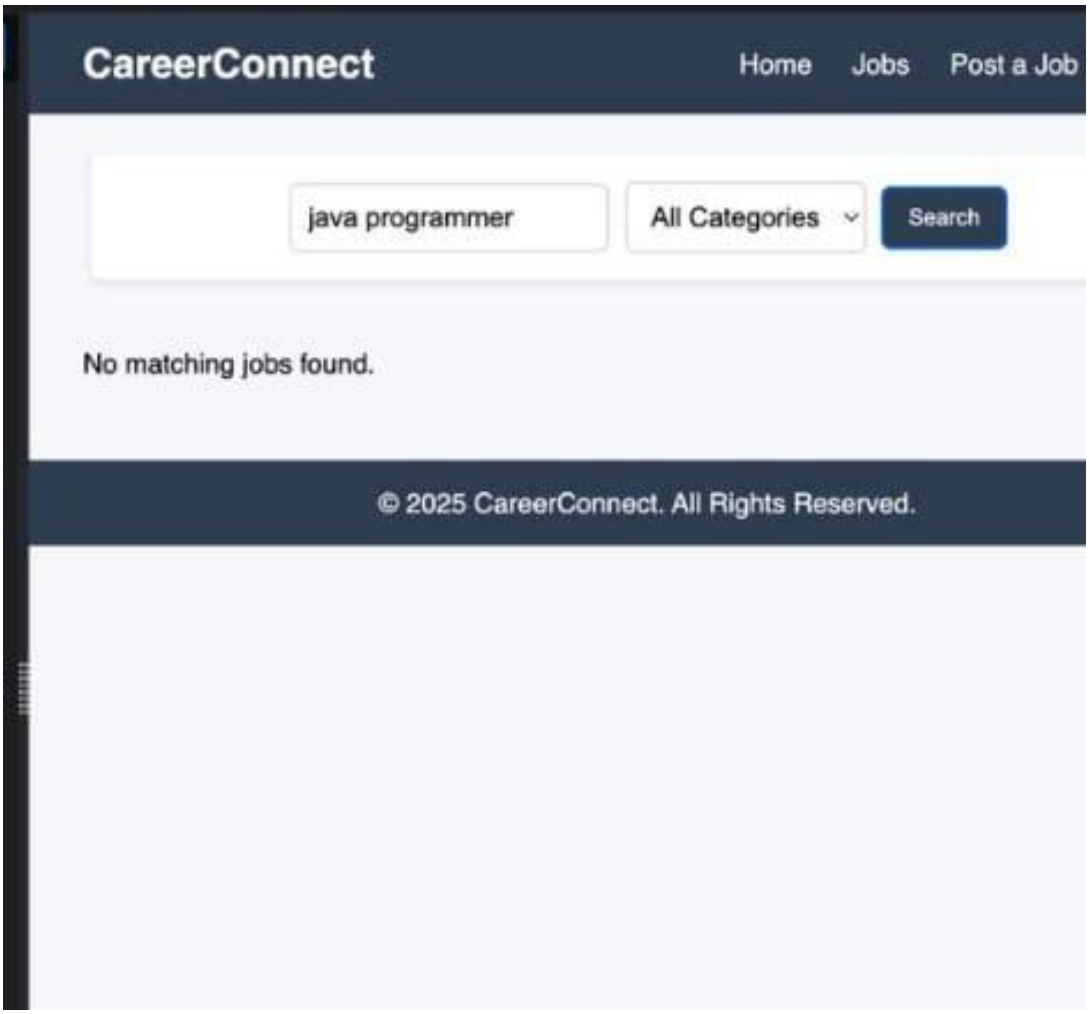
<!-- ===== JOB LISTINGS ===== -->
<section class="job-listings" id="jobContainer"></section>
<footer>
© 2025 CareerConnect. All Rights Reserved.
</footer>
<script>
const jobs = [
{ title: "Frontend Developer", company: "TechNova", category: "Tech", loc
{ title: "Graphic Designer", company: "Designify", category: "Design", loc
{ title: "Product Manager", company: "InnovateHub", category: "Management'
{ title: "Digital Marketer", company: "BrandBoost", category: "Marketing",
{ title: "Backend Engineer", company: "CloudSync", category: "Tech", locat
];

CHAPTER 6

RESULTS




Search jobs...
All Categories
Frontend Developer
TechNova - Remote
JavaScript
React
Apply Now
Graphic Designe
Designify - New York,
Figma
Photoshop
Apply Now
Product Manager
InnovateHub - London, UK
Leadership
Agile
Apply Now
Digital Marketer
BrandBoost - Toronto
SEO
Content
Apply Now
Backend Engineer
CloudSync - Berlin, Germany
Node.js
AWS
Apply Now


CareerConnect
Home
Jobs
Post a Job
java programmer
All Categories
Search
No matching jobs found.
© 2025 CareerConnect. All Rights Reserved.

CHAPTER 7

CONCLUSION

The CareerConnect website is a fully functional, client-side job-listing platform developed using HTML, CSS, and JavaScript. It successfully provides an intuitive and responsive interface for users to search, filter, and view job opportunities. The dynamic rendering of job cards, real-time filtering, and interactive elements enhance user experience and make job discovery simple and efficient.

The modular structure and organized code allow for easy scalability and future enhancements, such as backend integration, user authentication, and real-time job posting. Overall, CareerConnect demonstrates key web development concepts, including responsive design, DOM manipulation, and client-side data management, providing a strong foundation for building a complete recruitment platform.

REFERENCES

MDN Web Docs – HTML, CSS, JavaScript: https://developer.mozilla.org

W3Schools – HTML, CSS, JavaScript: https://www.w3schools.com

Duckett, J., HTML & CSS: Design and Build Websites, Wiley, 2011.

Duckett, J., JavaScript & JQuery: Interactive Front-End Web Development, Wiley, 2014.

Stack Overflow – https://stackoverflow.com

CSS-Tricks – https://css-tricks.com