# 

## 634675027549425435_sit

## Siliguri Institute of Technology

Salbari, PO: Sukna Darjeeling: - 734009

# Accessible Job Portal for People with Disabilities

## PROJECT SYNOPSIS

PROJECT – II  
PROJ-CS781

## Bachelor of Technology

Computer Science & Engineering

## SUBMITTED BY

CSE\_PROJ\_2025\_<Gr-Id>

**Group Members:**

|  |  |  |
| --- | --- | --- |
| **Student Names** | **University Roll No.** | **Registration No.** |
| SHUBHAM BHOWMIK | 11900121120 | 211190100110104 |
| DEBOJYOTI SEN | 11900121124 | 211190100110076 |
| SOUMYADEEP MUKHERJEE | 11900121126 | 211190100110075 |
| SUBHRAJIT BAGCHI | 11900121127 | 211190100110080 |

## Topic of the Project

Accessible Job Portal for People with Disabilities

## Project Category

Web Development

## INDEX

1. Statement About the Problem ...................................................................................................................................................... 3

2.Objective and Scope of the Project ................................................................................................................................................ 4

3. Methodology Adopted ....................................................................................................................................................................... 6

4. Tools and Platforms ............................................................................................................................................................................ 7

5. Algorithms to Be Implemented ..................................................................................................................................................... 8

6.Conclusion ............................................................................................................................................................................................. 11

7.Bibliography/Referencing ............................................................................................................................................................. 12

## 

## Statement about the Problem

Individuals with disabilities often face significant challenges when navigating traditional job portals, which are not designed with inclusivity and accessibility in mind. These challenges stem from multiple factors:

**Lack of Accessibility Features:**

Most existing job portals fail to comply with accessibility standards such as WCAG, making them difficult or impossible to use for individuals with visual, auditory, motor, or cognitive impairments.

Absence of essential features like screen reader compatibility, voice navigation, or adjustable contrast settings creates barriers for users with disabilities.

**Limited Customization Options:**

Users with specific needs often struggle with rigid interfaces that do not allow customization of font sizes, color contrasts, or layouts, resulting in a frustrating user experience.

**Insufficient Inclusive Job Listings:**

Employers rarely include information about accessibility features in their job postings, such as wheelchair access, remote work options, or accommodations for individuals with sensory impairments.

This lack of transparency discourages individuals with disabilities from applying for jobs, further widening the gap between employers and a diverse talent pool.

**Barriers to Communication:**

Traditional job platforms do not provide accessible communication tools such as live captioning, sign language interpretation, or voice-to-text features, limiting interactions between employers and job seekers with disabilities.

**Social Stigma and Exclusion:**

The systemic exclusion of individuals with disabilities from the workforce is exacerbated by a lack of inclusive hiring practices and awareness among employers. Many talented individuals are overlooked due to misconceptions about their abilities or insufficient accommodations in the hiring process.

**Missed Opportunities for Employers:**

Employers miss out on a diverse and skilled talent pool due to the absence of accessible platforms that facilitate inclusive hiring practices. Companies aiming to promote diversity and inclusion often lack the tools to effectively engage with candidates with disabilities.

The absence of an inclusive and accessible job portal perpetuates these challenges, creating a significant gap in the job market. This project aims to address these issues by providing a platform that prioritizes accessibility, fosters inclusivity, and empowers both job seekers with disabilities and employers to connect in a meaningful and equitable way.

## Objective and Scope of the Project

### Objective:

The primary objective of the project is to design and develop an Accessible Job Portal for People with Disabilities, aiming to create a user-friendly and inclusive digital hiring platform. This project seeks to address the challenges faced by individuals with disabilities in accessing job opportunities by incorporating features that cater to their specific needs. The key goals include:

**Enhancing Accessibility:**

Ensure compliance with Web Content Accessibility Guidelines (WCAG 2.1) to create a platform that is usable by people with diverse abilities, including those with visual, auditory, motor, or cognitive impairments. Provide assistive features such as screen reader compatibility, voice-based navigation, keyboard shortcuts, and adjustable interface elements like font size, color contrast, and layout customization.

**Promoting Equal Opportunities:**

Foster inclusivity in the job market by bridging the gap between employers and job seekers with disabilities. Encourage employers to include accessibility-specific metadata in job postings, allowing candidates to filter and apply for jobs that match their requirements.

**Improving User Experience:**

Develop an intuitive and seamless user interface that enables job seekers to easily search, filter, and apply for jobs. Provide tools for employers to efficiently post job openings, manage applications, and communicate with candidates in an inclusive manner.

**Facilitating Communication:**

Enable better interaction between job seekers and employers through accessible messaging systems, video interview tools with live captioning or sign language interpreter options, and feedback mechanisms.

**Empowering Stakeholders:**

Equip job seekers with features to create professional profiles, upload resumes, and track application statuses effortlessly. Offer employers insights into inclusivity practices and tools to reach a broader, diverse talent pool.

**Leveraging Technology:**

Integrate cutting-edge tools such as AI-driven resume matching, accessible search algorithms, and sentiment analysis for feedback. Incorporate voice-to-text and text-to-speech APIs to enhance usability for individuals with disabilities.

**Promoting Social Impact:**

Address the systemic barriers that prevent individuals with disabilities from accessing meaningful employment. Contribute to building a more inclusive society by encouraging diversity in the workplace and raising awareness about accessibility.

By achieving these objectives, the project not only demonstrates technical proficiency in web development and accessibility compliance but also highlights a commitment to social responsibility and innovation.

### Scope:

The project encompasses the following key areas:

**User Accessibility:**

Integrate assistive technologies such as screen reader compatibility, voice navigation, and keyboard shortcuts to cater to visually or physically impaired users.

Provide adjustable interface elements like font size, color contrast, and layout for users with specific accessibility needs.

**Job Filtering and Search:**

Offer advanced filters that allow users to find jobs based on accessibility features provided by employers (e.g., wheelchair access, remote work options).

Implement an intelligent search algorithm to prioritize job listings based on user preferences and accessibility requirements.

**Separate Dashboards for Stakeholders:**

For Job Seekers: Enable profile creation, resume uploads, and application tracking with a focus on accessibility.

For Employers: Facilitate job postings with accessibility-specific metadata and tools to manage applications efficiently.

**Inclusivity in Hiring:**

Incorporate features to promote better communication between employers and job seekers, including accessible messaging tools and live interview scheduling with accessibility options.

**Technology Integration:**

Leverage cutting-edge tools like voice-to-text and text-to-speech APIs to simplify navigation and interaction for all users.

Conduct sentiment analysis on feedback to ensure continuous improvement and inclusivity.

**Global Standards Compliance:**

Ensure adherence to WCAG 2.1 guidelines for accessibility and inclusivity.Conduct rigorous testing using tools like Lighthouse and Axe to maintain high standards of usability for users with disabilities. By focusing on these areas, the project aims to create a robust and scalable solution that bridges the gap between employers and individuals with disabilities, fostering a more inclusive hiring ecosystem.

## 

## Methodology Adopted

## The development of the Accessible Job Portal for People with Disabilities follows a structured methodology to ensure efficiency, accessibility compliance, and user satisfaction. The key phases include:

## Requirement Analysis:

## Conduct in-depth research on accessibility standards such as WCAG 2.1 and ARIA (Accessible Rich Internet Applications) guidelines to identify best practices. Gather feedback from potential users, including individuals with disabilities, to understand their specific needs and challenges when navigating existing job portals. Identify technical requirements for implementing assistive technologies such as screen readers, voice navigation, and keyboard accessibility.

## System Design:

## Create user-friendly layouts and workflows tailored for job seekers and employers. Develop wireframes and prototypes to visualize the user interface and ensure alignment with accessibility standards. Plan the architecture of the system, including frontend, backend, and database components, ensuring scalability and flexibility for future enhancements.

## Development:

## Frontend Development: Build a responsive and dynamic user interface using React.js, ensuring compatibility with assistive technologies and support for customizable UI elements like font size and contrast adjustments. Backend Development: Develop server-side logic using Node.js and Express.js to handle user authentication, job postings, and data processing.

## Database Integration:

## Use MongoDB to store and manage user profiles, job listings, and application data efficiently. Accessibility Features: Integrate tools such as voice-to-text APIs, text-to-speech APIs, and ARIA attributes to enhance accessibility.

## Algorithm Implementation:

## Accessible Search Algorithm: Implement a matching algorithm to filter job listings based on user preferences for accessibility features (e.g., wheelchair access, remote work).

## Resume Matching Algorithm: Develop an AI-driven system using cosine similarity to rank job recommendations based on the candidate’s skills and job requirements.

## Accessibility Compliance Checker: Create an algorithm to evaluate job postings for adherence to accessibility standards and flag missing features.

## Testing:

## Conduct rigorous testing using accessibility auditing tools like Lighthouse, Axe, and manual testing with screen readers. Involve real users, including individuals with disabilities, to provide feedback on usability, functionality, and accessibility. Perform stress testing to ensure the system can handle high traffic and large datasets without performance degradation.

## Deployment:

## Host the platform on a reliable cloud service such as AWS or Microsoft Azure, ensuring scalability and high availability. Configure Continuous Integration/Continuous Deployment (CI/CD) pipelines for automated updates and bug fixes. Monitor performance using analytics tools to track user engagement and identify areas for improvement.

## Post-Deployment Maintenance:

## Gather continuous feedback from users to identify and address issues or enhancements. Regularly update the platform to maintain compliance with evolving accessibility standards. Add new features, such as multilingual support and advanced analytics, based on user demand and market trends.

## This systematic methodology ensures the delivery of a robust, accessible, and user-friendly job portal that addresses the unique challenges faced by individuals with disabilities.

## Tools/Platforms

**Programming Languages:**

JavaScript: A versatile and widely-used programming language for creating interactive and dynamic web applications. Essential for implementing frontend interactivity and backend logic.

HTML & CSS: Standard languages for structuring and styling the web pages, ensuring compliance with accessibility standards.

**Frameworks:**

React.js: A popular JavaScript library for building fast and dynamic user interfaces. Its component-based architecture ensures modularity and reusability.

Node.js: A runtime environment that enables server-side scripting using JavaScript, ensuring scalability and high performance.

Express.js: A web application framework for Node.js that simplifies the development of robust APIs and server-side logic.

**Database:**

MongoDB: A NoSQL database known for its flexibility and scalability, ideal for storing user profiles, job listings, and application data.

**Testing Tools:**

Lighthouse: A tool for auditing web applications, focusing on performance, accessibility, and SEO.

Axe: A comprehensive accessibility testing tool that identifies and resolves WCAG compliance issues.

Screen Readers: Assistive technologies that enable visually impaired users to navigate the platform seamlessly.

**Deployment Platforms:**

AWS (Amazon Web Services): A cloud computing platform providing reliable hosting services with scalability.

Microsoft Azure: Another robust cloud platform that ensures high availability and global reach.

## Algorithms to be Implemented

### Accessible Search Algorithm

#### Input:

- User preferences (e.g., wheelchair access, sign language interpreter, remote work).

- Job listings with metadata on accessibility features.

#### Processing:

- Loop through all job listings.

- Check each listing for matching accessibility features with user preferences.

- Assign a priority score based on the number of matching features.

- Sort the listings by the priority score in descending order.

***Algorithm (Pseudocode)***

def accessible\_search(user\_preferences, job\_listings):

results = []

for job in job\_listings:

score = 0

for feature in user\_preferences:

if feature in job['accessibility\_features']:

score += 1

results.append((job, score))

results.sort(key=lambda x: x[1], reverse=True)

return [job for job, score in results]

#### Output:

A filtered and sorted list of job listings, with higher priority scores shown first.

**Resume Matching Algorithm:**

#### Input:

- Candidate’s resume (text format).

- Job descriptions with required skills.

#### Processing:

- Extract keywords from the resume and job descriptions.

- Use cosine similarity to measure the match between the resume and job requirements.

- Rank jobs based on similarity scores.

***Algorithm (Pseudocode):***

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.metrics.pairwise import cosine\_similarity

def match\_resume(resume, job\_descriptions):

vectorizer = CountVectorizer().fit\_transform([resume] + job\_descriptions)

vectors = vectorizer.toarray()

resume\_vector = vectors[0]

job\_vectors = vectors[1:]

scores = cosine\_similarity([resume\_vector], job\_vectors).flatten()

matched\_jobs = sorted(zip(job\_descriptions, scores), key=lambda x: x[1], reverse=True)

return matched\_jobs[:5]

#### Output:

A list of job recommendations with the highest similarity scores.

### Accessibility Compliance Checker

#### Input:

- Accessibility features provided by the employer.

- List of required accessibility standards (e.g., WCAG).

#### Processing:

- Compare the provided features with the required standards.

- Generate a compliance report and flag missing features.

***Algorithm (Pseudocode):***

def check\_accessibility\_compliance(provided\_features, required\_standards):

missing\_features = [feature for feature in required\_standards if feature not in provided\_features]

if missing\_features:

return f"Non-compliant. Missing features: {missing\_features}"

return "Compliant"

#### Output:

Approval or rejection of the job listing based on compliance.

### Voice Navigation Algorithm

#### Input:

- User voice commands (e.g., 'Search jobs,' 'Apply to job 3').

Processing:

- Use a Speech-to-Text API to convert voice commands into text.

- Parse the command to determine the action (e.g., search, filter, apply).

***Algorithm (Pseudocode):***

def voice\_navigation(command):

action = parse\_command(command) # Extract action and parameters

if action == "search":

return perform\_search(parameters)

elif action == "apply":

return apply\_to\_job(parameters['job\_id'])

else:

return "Command not recognized"

#### Output:

Execution of the desired action and feedback via Text-to-Speech (TTS).

### Sentiment Analysis for Feedback System

### *Input:*

- User feedback (text format).

#### Processing:

- Use an NLP library (e.g., TextBlob or VADER) to classify feedback as positive, neutral, or negative.

- Count the feedback in each sentiment category.

***Algorithm (Pseudocode):***

from textblob import TextBlob

def analyze\_feedback(feedback\_list):

sentiments = {"positive": 0, "neutral": 0, "negative": 0}

for feedback in feedback\_list:

sentiment = TextBlob(feedback).sentiment.polarity

if sentiment > 0:

sentiments["positive"] += 1

elif sentiment == 0:

sentiments["neutral"] += 1

else:

sentiments["negative"] += 1

return sentiments

#### Output:

A summary report of user sentiment trends.

## 

## 

## Conclusion

The Accessible Job Portal for People with Disabilities represents a groundbreaking initiative in the domain of accessible technology and inclusive design. By leveraging cutting-edge tools and adhering to global accessibility standards like WCAG, this project bridges a significant gap in the job market, empowering individuals with disabilities to overcome traditional barriers to employment.

The portal not only simplifies the job search process for people with disabilities but also fosters a culture of inclusivity and diversity in workplaces. Through features such as AI-driven resume matching, accessible search filters, and assistive technologies like voice navigation and screen reader support, the project demonstrates how technology can be harnessed to create meaningful social impact.

Moreover, the dual-dashboard system for job seekers and employers ensures that both parties benefit from a seamless and user-friendly experience. Job seekers are empowered to highlight their skills and connect with inclusive employers, while companies gain the tools to foster diversity and comply with accessibility standards.

Beyond its technical achievements, the project serves as a testament to the potential of technology in addressing societal challenges. By prioritizing accessibility and inclusivity, it creates a platform that reflects the values of equity and equal opportunity.

The project’s scalability and adaptability ensure that it can be expanded to include additional features, such as multilingual support, advanced analytics, and partnerships with organizations focused on disability advocacy. Its innovative approach and impactful outcomes position it as a model for future technological solutions aimed at bridging societal gaps.

In conclusion, the Accessible Job Portal is not just a web application but a step forward in creating a more equitable and inclusive digital ecosystem. It sets a precedent for how technology can be used to empower marginalized communities, making a tangible difference in the lives of individuals with disabilities and paving the way for a more inclusive job market.

## 

## Bibliography/Referencing

Web Content Accessibility Guidelines (WCAG) 2.1 – W3C:

Comprehensive guidelines for making web content more accessible to people with disabilities.

Link: <https://www.w3.org/WAI/standards-guidelines/wcag/>

React.js Documentation:

Official documentation for React.js, a popular frontend library used for building user interfaces.

Link: <https://reactjs.org/>

Node.js Documentation:

Official documentation for Node.js, a JavaScript runtime environment for server-side development.

Link: <https://nodejs.org/en/docs/>

MongoDB Documentation:

Resources and guides for working with MongoDB, a NoSQL database used for managing data.

Link: <https://www.mongodb.com/docs/>

Axe Accessibility Tool:

Accessibility testing tool for identifying and resolving WCAG compliance issues.

Link: <https://www.deque.com/axe/>

Lighthouse Accessibility Auditing Tool:

A tool for auditing the performance, accessibility, and SEO of web applications.

Link: <https://developers.google.com/web/tools/lighthouse>

ARIA (Accessible Rich Internet Applications) Guidelines:

Standards for enhancing web application accessibility.

Link: <https://www.w3.org/WAI/ARIA/>

Inclusive Design Principles:

A set of principles to ensure inclusivity in design practices.

Link: <https://inclusivedesignprinciples.org/>

Microsoft Azure Documentation:

Resources for hosting and scaling applications on Azure cloud services.

Link:<https://learn.microsoft.com/en-us/azure/>

AWS Documentation:

Official guides and tutorials for using Amazon Web Services (AWS) for cloud hosting.

Link: <https://aws.amazon.com/documentation/>

TextBlob Documentation:

A Python library for processing textual data and performing sentiment analysis.

Link: <https://textblob.readthedocs.io/en/dev/>

Articles on Inclusive Web Design:

Research articles and case studies highlighting best practices for designing accessible web applications.

Third-Party APIs for Accessibility:

Documentation for integrating APIs like Google Speech-to-Text and Text-to-Speech for enhancing usability.