Hard bound cover

JOB APPLICATION PORTAL

A project report

Submitted in partial fulfilment of the requirements for the award of degree of

Name of Programme

Bachelors in technology

Submitted to

LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB

SUBMITTED BY

Name of student

Akash Nagineni

Registration Number

12008727

Name of Supervisor

UID of Supervisor

Annexure-V

Declaration by student

To whom so ever it may concern

I, **Akash Nagineni, 12008727,** hereby declare that the work done by me on "**Job Application Portal**" under the supervision of **Dr.Navneet kaur (24950)**, **Assistant professor,** Lovely professional University, Phagwara, Punjab, is a record of original work for the partial fulfilment of the requirements for the award of the degree, **Bachelor of Technology.**

Akash Nagineni (12008727)

Signature of the student

Dated: 21-04-2024

Annexure-VI

Declaration by the supervisor

To whom so ever it may concern

This is to certify that **Akash Nagineni** , **12008727** from Lovely Professional University, Phagwara, Punjab, has worked on "**Job Application Portal**" under my supervision from. It is further stated that the work carried out by the student is a record of original work to the best of my knowledge for the partial fulfilment of the requirements for the award of the degree, degree name.

Name of Supervisor

UID of Supervisor

Signature of Supervisor

Introduction

The Software Requirement Specification (SRS) document provides a comprehensive overview of the job application portal to be developed using Django framework. This document serves as a guide for software engineers, outlining the detailed requirements necessary to design and implement the software product effectively. The job application portal aims to revolutionize the recruitment process by providing a centralized platform for job seekers and employers to connect seamlessly. Through this portal, job seekers can explore relevant job opportunities, submit applications, and manage their profiles, while employers can post job listings, review applications, and communicate with candidates efficiently.

The SRS document encompasses all essential aspects of the job application portal, including functional requirements, non-functional requirements, user interface specifications, system constraints, and external interfaces. It is structured to provide clear and concise information to software engineers, enabling them to develop a robust and user-friendly software solution that meets the needs of both job seekers and employers. Additionally, the document will undergo continuous refinement and validation throughout the software development lifecycle to ensure alignment with stakeholder expectations and project objectives.

1.1 Purpose:

The purpose of this Software Requirement Specification (SRS) document is to define the functional and non-functional requirements of the job application portal developed using Django framework. It serves as a detailed guide for software engineers and developers, providing them with a clear understanding of the project scope, objectives, and specific features to be implemented. Additionally, the SRS document serves as a reference for stakeholders, including project managers, quality assurance teams, and end-users, to ensure alignment with the desired outcomes and expectations.

The intended audience for this SRS document includes:

1. Software Engineers and Developers: Responsible for designing, implementing, and testing the job application portal based on the outlined requirements and specifications.

2. Project Managers: Overseeing the project progress, resource allocation, and ensuring that the development aligns with the defined requirements and timelines.

3. Quality Assurance Teams: Responsible for validating the functionality and performance of the job application portal through rigorous testing processes.

4. Stakeholders: Including executives, investors, and other relevant parties interested in understanding the project scope, objectives, and deliverables.

5. End-users: Both job seekers and employers who will interact with the job application portal, ensuring that their needs and expectations are addressed effectively.

1.2 Scope:

(1) The software product to be produced is a job application portal developed using the Django framework.

(2) The job application portal will:

- Provide a user-friendly interface for job seekers to search for relevant job opportunities, submit applications, and manage their profiles.
- Offer tools for employers to post job listings, review candidate applications.
- Include features such as advanced search functionality, resume uploading, application tracking.
- Ensure data security and privacy by implementing appropriate encryption and access control measures.
- Allow for scalability and flexibility to accommodate future enhancements and integration with external systems.

The job application portal will not:

- Guarantee job placement or employment outcomes for job seekers.
- Provide recruitment services beyond facilitating the connection between job seekers and employers.
- Offer automated decision-making processes for hiring or screening candidates.

(3) Application of the software:

(a) The job application portal aims to achieve the following benefits, objectives, and goals:

- Facilitate efficient and transparent recruitment processes for both job seekers and employers.
- Provide a centralized platform for job search, application submission, and candidate evaluation.
- Enable employers to identify and engage with top talent effectively, reducing time-to-hire and cost-per-hire metrics.
- Empower job seekers to showcase their skills and experiences to potential employers in a user-friendly manner.
- Enhance accessibility and convenience by offering mobile responsiveness and intuitive navigation.
- Ensure compliance with data privacy regulations and industry standards to protect user information.

(b) This scope is consistent with the objectives outlined in higher-level specifications, such as the System Requirement Specification, ensuring alignment with project goals and stakeholder expectations.

1.3 Definitions, Acronyms, and Abbreviation

Definitions:

1. **J**ob Application Portal: A web application designed to facilitate the process of job applications, typically allowing users to search for job listings, submit resumes, and communicate with employers.

2. Django: A high-level Python web framework that encourages rapid development and clean, pragmatic design.
3. User: A person who interacts with the job application portal, including job seekers and employers.
4. Administrator: A user role with elevated privileges, responsible for managing users, job listings, and other aspects of the portal.
5. Authentication: The process of verifying the identity of users accessing the job application portal, typically through login credentials such as username and password.
6. CRUD Operations: Create, Read, Update, and Delete operations, representing the basic functions for managing data in a database.

Acronyms and Abbreviations:

1. SRS: Software Requirements Specification
2. CRUD: Create, Read, Update, Delete
3. DB: Database
4. CSS: Cascading Style Sheets
5. HTML: Hypertext Markup Language
6. ORM: Object-Relational Mapping
7. HTTP: Hypertext Transfer Protocol

1.4 References

1.Django Documentation:

- Title: Django Documentation
- Publishing Organization: Django Software Foundation
- Sources: Available online at https://docs.djangoproject.com/en/5.0/

2.GeeksforGeeks Django Tutorials:

- Title: Django Tutorials on GeeksforGeeks
- Publishing Organization: GeeksforGeeks
- Sources: Available online at https://www.geeksforgeeks.org/django-tutorial/.

3.Tutorialspoint Django Tutorial:

- Title: Django Tutorial on Tutorialspoint
- Publishing Organization: Tutorialspoint
- Sources: Available online at https://www.tutorialspoint.com/django/index.htm.

4.W3Schools Django Tutorial:

- Title: Django Tutorial on W3Schools
- Publishing Organization: W3Schools
- Sources: Available online at https://www.w3schools.com/django/default.asp.

1.5 Overview:

(1) The rest of the Software Requirement Specification (SRS) contains detailed information about the job application portal developed using Django framework. It includes comprehensive documentation of the functional and non-functional requirements, user interface specifications, system constraints, and external interfaces necessary for designing and implementing the

software product effectively. Additionally, the SRS outlines the scope, purpose, and application of the job application portal, providing a holistic view of the project for stakeholders and software engineers alike.

(2) The SRS is organized into several sections, each addressing specific aspects of the job application portal:

- Introduction: Provides an overview of the SRS document and its intended audience.
- Purpose: Defines the purpose of the SRS and identifies its intended audience.
- Scope: Describes the software product(s) to be produced, including what it will and will not do, as well as its application and objectives.
- Definitions, Acronyms, and Abbreviations: Lists and defines any technical terms, acronyms, or abbreviations used throughout the document.
- References: Includes any external documents or resources referenced in the SRS.
- Functional Requirements: Details the functional specifications of the job application portal, including user interactions, system behaviours, and data processing.
- Non-Functional Requirements: Outlines the non-functional specifications, such as performance, security, usability, and scalability considerations.
- User Interface Specifications: Describes the visual and interactive aspects of the user interface, including layout, navigation, and design elements.
- System Constraints: Identifies any limitations or constraints imposed by the system architecture, hardware, or software environment.
- External Interfaces: Specifies any external systems, APIs, or databases that the job application portal will interact with.
- Appendices: Includes additional supplementary information, such as mock-ups, diagrams, or supporting documentation.

By organizing the SRS in this structured manner, it provides a comprehensive and systematic guide for software engineers to design, develop, and test the job application portal effectively, ensuring alignment with stakeholder requirements and project objectives.

2. General Description:

The job application portal developed using Django framework is a web-based platform designed to streamline the recruitment process for both job seekers and employers. This section of the Software Requirement Specification (SRS) provides a high-level overview of the general factors that influence the product and its requirements, offering context and clarity to stakeholders and software engineers involved in the project.

1.Purpose: The primary purpose of the job application portal is to serve as a centralized hub where job seekers can search for job opportunities and submit applications, while employers can post job listings, review candidate profiles, and manage the hiring process efficiently. By facilitating seamless communication and interaction between job seekers and employers, the portal aims to simplify recruitment efforts and enhance the overall experience for all stakeholders.

2.Target Audience: The target audience for the job application portal includes job seekers actively seeking employment opportunities across various industries and sectors, as well as employers seeking to fill vacant positions within their organizations. Additionally, the portal may cater to recruiters, HR professionals, and hiring managers responsible for managing recruitment processes and evaluating candidates.

3.Functionality: The job application portal will offer a range of functionality to meet the needs of its users. This includes features such as advanced job search filters, resume uploading, application tracking, communication tools, and notification alerts. The portal will prioritize user experience and accessibility, ensuring intuitive navigation and responsive design across different devices and screen sizes.

4.Security and Privacy: Given the sensitive nature of personal and professional information exchanged on the portal, security and privacy considerations are paramount. The portal will implement robust security measures, including encryption protocols, secure login authentication, and access control mechanisms, to safeguard user data and prevent unauthorized access or data breaches.

5.Scalability and Flexibility: As the job application portal is expected to cater to a growing user base and evolving industry needs, scalability and flexibility are essential considerations. The portal will be designed with modular architecture and scalable infrastructure to accommodate increasing user traffic and future enhancements or integrations with external systems.

2.1 Product Perspective

The job application portal will be designed to offer a seamless and intuitive user experience for both job seekers and employers. Key features will include robust search functionality with advanced filtering options to match candidates with relevant job postings efficiently. Job seekers will have the ability to create detailed profiles, upload resumes, and track application statuses, while employers will benefit from tools for managing job listings, screening applicants, and scheduling interviews.

The portal will prioritize scalability and flexibility, allowing for future enhancements and integration with emerging technologies to adapt to the evolving needs of the recruitment landscape. Overall, the job application portal will serve as a comprehensive solution that

simplifies the recruitment process for all stakeholders involved, ultimately driving greater efficiency and success in matching candidates with their ideal employment opportunities.

2.2 Product Functions

The job application portal will provide the following core functions:

1.User Registration and Authentication:

- Allow users to create accounts by providing necessary information.
- Authenticate users securely during login using credentials.

2.Job Seeker Features:

- Search for job listings based on various criteria such as job title, location, and category.
- View detailed information about job listings including descriptions, requirements, and application deadlines.
- Submit job applications by uploading resumes and cover letters.

3.Employer Features:

- Post job listings with detailed descriptions, requirements, and application instructions.
- Manage job listings including editing, updating, and removing listings as necessary.
- Communicate with job seekers through messaging or other communication channels.

4.Administrator Functions:

- Manage user accounts, including creating, editing, and deactivating accounts.

2.3 User Characteristics

The job application portal is designed to accommodate the following types of users, each with distinct characteristics that may influence specific requirements:

Job Seekers:

- Search for jobs with extensive filters (location, t)
- Apply for unlimited jobs across different categories.
- Manage profiles with experience, education, skills, and resume.

Employers:

- Posts jobs on different categories
- Monitor job posting and applications through the dashboard.
- Manage job posting and users profiles.

Administrators:

- CRUD operations for users and Employers.
- Dynamic UI management.
- Website metadata control.

2.4 General Constraints

The development of the job application portal is constrained by:

1.Technology Stack:

- Developed using Django web framework in Python.
- Integration with HTML, CSS, and JavaScript for frontend.

2.Database:

- Relational database management system (RDBMS) like PostgreSQL or MySQL.

3.Scalability:

- Architecture must accommodate future scalability needs.
- Consideration of cloud infrastructure for scalability.

4.Security:

- Adherence to industry-standard security practices.
- Encryption, secure authentication, and regular security audits.

5.Regulatory Compliance:

- Compliance with data protection and privacy laws.
- Adherence to industry-specific regulations.

6.User Experience:

- Responsive and intuitive user interface design.
- Optimization of page load times and system performance.

7.Resource Allocation:

- Sufficient resources including time, budget, and personnel.
- Optimization of resource utilization.

8.Third-Party Integration:

- Integration with third-party services and APIs.
- Consideration of third-party limitations and constraints.

2.5 Assumptions and Dependencies

The requirements stated in this SRS are based on the following assumptions and dependencies:

1.Availability of Required Hardware and Software:

- Assumption: The hardware and software environments necessary for deploying the job application portal will be available as specified.
- Dependency: Any changes or discrepancies in the availability of hardware or software may require adjustments to the system requirements and configurations.

2.Internet Connectivity:

- Assumption: Users will have access to stable internet connectivity to interact with the job application portal.
- Dependency: Reliability of the portal's functionality is dependent on uninterrupted internet access for users.

3.Compliance with Third-Party APIs and Services:

- Assumption: Integration with third-party services and APIs will function as expected.
- Dependency: Any changes to the functionality or availability of third-party APIs and services may impact the functionality of the job application portal and require corresponding modifications.

4.User Data Privacy and Security:

- Assumption: Adequate measures will be in place to ensure the privacy and security of user data collected and processed by the job application portal.
- Dependency: Adherence to data protection regulations and industry standards is necessary to maintain user trust and legal compliance.

5.Stakeholder Collaboration and Feedback:

- Assumption: Stakeholders will actively participate in the requirements gathering process and provide timely feedback during development.
- Dependency: The accuracy and completeness of the SRS rely on effective communication and collaboration with stakeholders throughout the project lifecycle.

Here's how you can structure these sections for your Django job application portal:

3.1 External Interface Requirements

3.1.1 User Interfaces

- The user interface of the job application portal should be intuitive and easy to navigate for both job seekers and employers.
- It should support various functionalities such as user registration, job search, application submission, and job listing management.

3.1.2 Hardware Interfaces

- The job application portal should be compatible with standard hardware configurations commonly used for web browsing, including desktop computers, laptops, tablets, and smartphones.
- No specific hardware interfaces are required.

3.1.3 Software Interfaces

- Integration with a relational database management system (RDBMS) such as PostgreSQL or MySQL for data storage.
- Utilization of Django's built-in authentication system for user authentication and authorization.

3.1.4 Communications Interfaces

- The portal should support communication between users (e.g., messaging between employers and job seekers).
- Integration with email services for notifications and communication with users.

3.2 Functional Requirements

3.2.1 Job Listing Management

- **Introduction:** Employers should be able to post, edit, and remove job listings.
- **Inputs:** Job details including title, description, requirements, and application instructions.
- **Processing:** Validation of inputs, storing job listings in the database.
- **Outputs:** Published job listings visible to job seekers.
- **Error Handling:** Notify users of errors during input validation or database operations.

3.2.2 Job Search and Application

- **Introduction:** Job seekers should be able to search for job listings and submit applications.
- **Inputs:** Search criteria (e.g., job title, location), application details (resume, cover letter).
- **Processing:** Matching search criteria to job listings, storing application details.
- **Outputs:** Display search results, confirmations for successful application submissions.
- **Error Handling:** Notify users of search query errors or application submission failures.

3.5 Non-Functional Requirements

3.5.1 Performance

- Job search results should be returned within 2 seconds.
- Application submission should not take more than 5 seconds to process.

3.5.2 Reliability

- The job application portal should have a minimum uptime of 99.9%.
- It should be capable of handling concurrent user interactions without system crashes.

3.5.3 Availability

- The portal should be accessible 24/7 except during scheduled maintenance windows.

3.5.4 Security

- User data should be encrypted during transmission and storage.

- Access to sensitive data and functionalities should be restricted based on user roles and permissions.

3.5.5 Maintainability

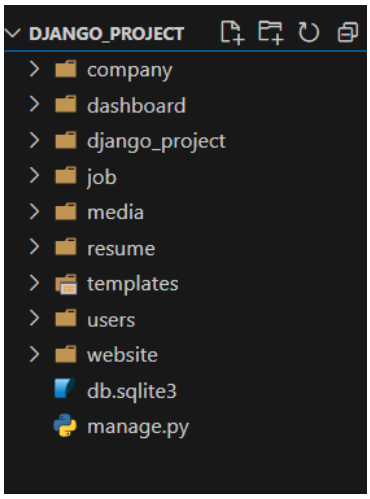
- Code should be well-organized and documented to facilitate future updates and maintenance.
- Regular code reviews and testing should be conducted to identify and address issues promptly.

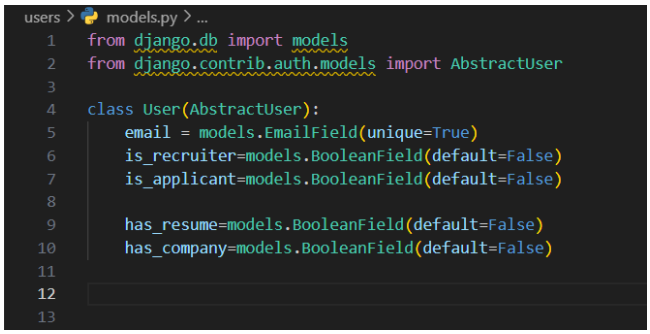
3.5.6 Portability

- The job application portal should be compatible with major web browsers (Chrome, Firefox, Safari) and operating systems (Windows, macOS, Linux).

3.7 Design Constraints

- Compliance with Django's MVC (Model-View-Controller) architecture.
- Adherence to Django's coding standards and best practices.
- Utilization of Django's ORM (Object-Relational Mapping) for database interactions.





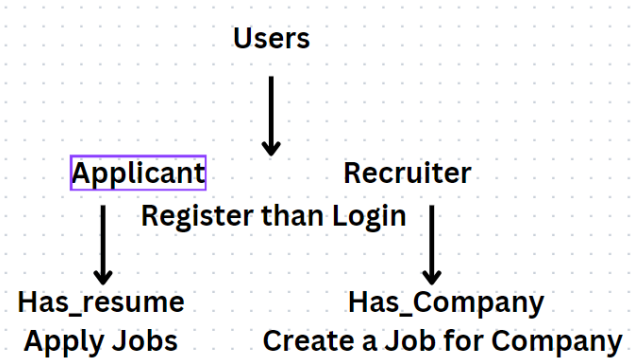
company > models.py > Company

```python
from django.db import models
from users.models import User

class Company(models.Model):
    user=models.OneToOneField(User,on_delete=models.CASCADE)
    name=models.CharField(max_length=100,null=True,blank=True)
    est_date=models.PositiveIntegerField(null=True,blank=True)
    city=models.CharField(max_length=100,null=True,blank=True)
    state=models.CharField(max_length=100,null=True,blank=True)

    def __str__(self):
        return self.name
```

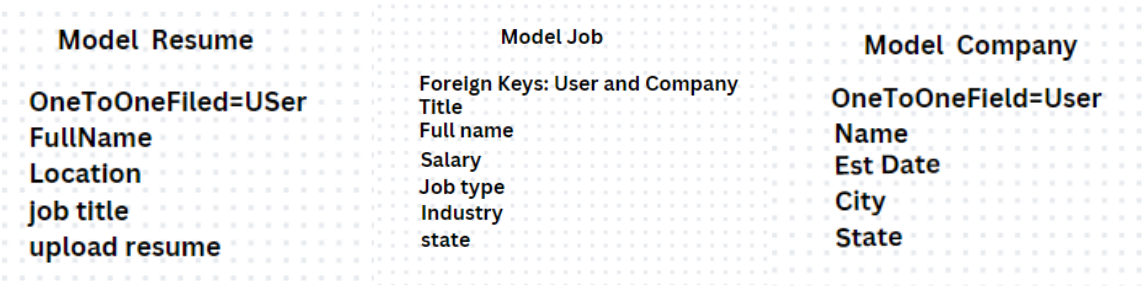
resume > models.py > Resume

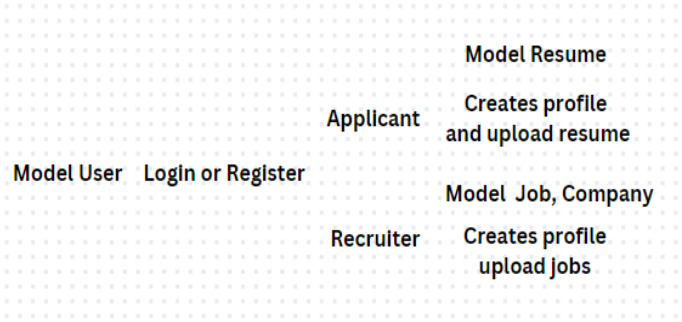
```python
from django.db import models
from users.models import User

class Resume(models.Model):
    user=models.OneToOneField(User,on_delete=models.CASCADE)
    first_name=models.CharField(max_length=100,null=True,blank=True)
    surname=models.CharField(max_length=100,null=True,blank=True)
    location=models.CharField(max_length=100,null=True,blank=True)
    job_title=models.CharField(max_length=100,null=True,blank=True)
    upload_resume=models.FileField(upload_to='resume',null=True,blank=True)

    #insert cv
    def __str__(self):
        return f'{self.first_name} {self.surname}'
```

```python
class Job(models.Model):
    job_type_choices=(
        ('Remote','Remote'),
        ('Onsite','Onsite'),
        ('Hybrid','Hybrid'),
    )
    user=models.ForeignKey(User,on_delete=models.CASCADE)
    company=models.ForeignKey(Company,on_delete=models.CASCADE)
    title=models.CharField(max_length=100)
    city=models.CharField(max_length=100)
    salary=models.PositiveBigIntegerField(default=35000)
    requirements=models.TextField()
    ideal_candidate=models.TextField()
    is_available=models.BooleanField(default=True)
    timestamp=models.DateTimeField(auto_now_add=True)
    industry=models.ForeignKey(Industry,on_delete=models.DO_NOTHING,null=True,blank=True)
    state=models.ForeignKey(State,on_delete=models.DO_NOTHING,null=True,blank=True)
    job_type=models.CharField(max_length=20,choices=job_type_choices,null=True, blank=True)

    def __str__(self):
        return self.title
```



Models:





Analysis Models for Django Job Application Portal

1. Use Case Diagram

- **Introduction:** The use case diagram for the Django job application portal illustrates the interactions between actors (job seekers, employers, administrators) and the system. It highlights the functionalities and features of the portal from a user's perspective.
- **Narrative Description:** Actors interact with the system through use cases such as "Search Job Listings," "Submit Application," "Post Job Listing," and "Manage User Account." These use cases represent the core functionalities of the portal as outlined in the SRS. The use case diagram aids in understanding how different actors engage with the system to achieve their goals, ensuring alignment with the specified requirements.

2. Sequence Diagram

- **Introduction:** Sequence diagrams for the Django job application portal illustrate the sequence of interactions between components such as views, models, and databases during specific scenarios or user actions.
- **Narrative Description:** For example, a sequence diagram depicting the process of job application submission would show interactions between the user interface, Django views handling form submissions, and database models for storing application data. It outlines the sequence of messages exchanged between these components, including user input, data validation, and database updates. This model helps validate requirements related to input processing, data flow, and system response times.

3. Entity-Relationship Diagram (ERD)

- **Introduction:** The ER diagram for the Django job application portal represents the structure of the database schema, including entities, attributes, and relationships between them.

- **Narrative Description:** Entities such as User, JobListing, Application, and Administrator are depicted, along with their attributes and relationships. For example, the ER diagram shows that a User can have multiple Applications, and a JobListing can have multiple Applications from different Users. It ensures that the database schema supports the data requirements specified in the SRS and facilitates efficient data management within the Django framework.

4. Class Diagram

- **Introduction:** Class diagrams for the Django job application portal illustrate the object-oriented design of the system, including classes, attributes, methods, and relationships.
- **Narrative Description:** Classes such as User, JobListing, Application, and Administrator are represented, along with their attributes and methods. Associations between classes, such as ownership of job listings by employers or relationships between users and applications, are depicted. This model provides a blueprint for implementing the system's functionality within the Django framework, ensuring that classes and their relationships align with the specified requirements in the SRS.

A. Appendices

A.1 Conceptual Documents

- This appendix includes conceptual documents such as system architecture diagrams, wireframes, and mockups. These documents provide a high-level overview of the system's design and functionality.

A.2 Marketing Materials

- This appendix contains marketing materials such as brochures, flyers, and promotional videos. These materials showcase the features and benefits of the job application portal to potential users and stakeholders.

A.3 Meeting Minutes

- This appendix includes minutes of meetings with the customer(s) and other stakeholders. It documents discussions, decisions, and action items related to the development of the job application portal, providing transparency and accountability throughout the project lifecycle.

A.4 User Feedback

- This appendix contains feedback collected from users during the testing and evaluation phases of the project. It includes user surveys, usability testing reports, and user interviews, highlighting areas for improvement and informing future iterations of the portal.

A.5 Technical Documentation

- This appendix includes technical documentation such as API specifications, database schemas, and deployment guides. It provides detailed information for developers,

system administrators, and other technical stakeholders involved in the implementation and maintenance of the job application portal.

GitHub Link:

Conclusion

The job application portal serves as a vital tool for connecting job seekers with employment opportunities and assisting employers in their recruitment efforts. Its intuitive interface and comprehensive features streamline the job search process for candidates while empowering employers to efficiently manage their recruitment pipeline. By providing a platform for seamless communication and interaction within the recruitment ecosystem, the portal plays a pivotal role in facilitating career advancement and organizational growth. However, like any evolving system, the portal requires constant refinement to meet the dynamic needs of its users effectively.

Addressing the identified areas for improvement and implementing the recommended enhancements can further enhance the portal's effectiveness, user satisfaction, and overall impact on the recruitment process. Refinements such as enhancing search algorithms for more relevant job listings and streamlining the application process for candidates can significantly improve user experience. Moreover, providing employers with advanced tools for candidate evaluation and tracking can boost efficiency and effectiveness in recruitment efforts. By embracing innovation and user-centric design principles, the portal can continue to evolve, ensuring its relevance and efficacy in connecting talent with opportunity in the ever-changing job market landscape.

PROJECT NAME	Job Application Portal		
JOB LOCATION	Pune		
EST. START DATE	1st MAR 2024	EST. FINISH DATE	22st APR 2024
PROJECT LEADER	Akash Nagineni	COMPANY	Jobify
CONTACT NAME	Nagarjuna	ADDRESS	Lovely Professional University
PHONE	9848780624		
EMAIL	nagarjuna@gmail.com		

SUMMARY	Job Application Portal Using Django
DESIRED OUTCOME	Need of Fully Functional Website along with backend
ACTION TO COMPLETION	Developed website using Django framework that fulfilled their needs.
BENEFITS OF PROJECT	Easy to manage and update in future with less to no expenditure.
PROJECTED SCHEDULE	Under one month
PROJECTED BUDGET	200-500 rupees
PROJECTED TEAM AND RESOURCE REQUIREMENTS	One person and a computer
PROPOSAL MAY BE WITHDRAWN IF NOT ACCEPTED BY DATE OF	22th APR 2024

ACCEPTANCE OF PROPOSAL			
AUTHORIZED CLIENT SIGNATURE	Sreenithin	DATE OF ACCEPTANCE	22rd APR 2024