Rafat Muskan Shaikh, 2023, 11:5
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

JobHub: A Django-Based Job Portal Connecting Job Seekers & Hirers

Rafat Muskan Shaikh, Moin Naik, Saquib Patel, Neeraj Kumar, Asst. Prof. Ramya Kanagaraj
Department of Computer Engineering,
Rizvi College of Engineering,
Mumbai, India

Abstract- JobHub is an innovative web application designed to bridge the gap between job seekers and hirers. This Django-based project employs HTML, Django templates, CSS, and JavaScript with an AJAX framework to create a seamless and user-friendly experience. The platform operates on a two-user model: "job seekers" and "hirers." Hirers can post job applications, but only if they complete their profiles to 100%. These job listings are stored in a SQLite3 database. Job seekers can view job postings and apply if their profiles are also at 100% completion. One of the key features of JobHub is real-time communication. When a job seeker applies for a job, the hirer receives an instant notification, allowing them to review and respond to each application individually. Job seekers are notified when they receive a response, ensuring a timely and efficient hiring process. To enhance user experience, JobHub incorporates various filters and a search bar, enabling users to find the most relevant job listings. Additionally, a sophisticated recommendation algorithm analyzes job seeker skills and matches them with the skills required for posted jobs, further increasing the chances of finding the perfect job fit. In summary, "JobHub" is a dynamic Django project that leverages modern web technologies to streamline the job search and hiring process, fostering connections between job seekers and hirers while providing valuable real-time notifications and intelligent job recommendations.

Keywords- Job Portal, Django Project, Job Recommendation, Job Postings, SQLite3.

I. INTRODUCTION

In an increasingly dynamic job market, connecting job seekers with suitable employment opportunities has become essential. To address this challenge, we present an innovative web application: "JobHub." This project is designed to facilitate seamless interactions between two key user groups: "Job Seekers" and "Hirers." JobHub leverages the power of Django, HTML, Django templates, CSS, JavaScript, and the Ajax framework to create a feature-rich platform that streamlines the job application process.

1. Key Features:

- **User Segmentation:** JobHub caters to two distinct user roles: "Job Seeker" and "Hirer." Each user type enjoys a tailored experience based on their needs and objectives.
- **Profile Completion:** Hirers can post job listings only after completing their profiles to 100%. This ensures the authenticity and credibility of job postings.
- **Job Posting:** Once hirers have completed their profiles, they can effortlessly post job openings. These listings are stored securely in the SQLite3 database for easy access and management.
- **Job Seeker Access:** Job seekers can browse and search for job listings based on various filters and a powerful search bar. This empowers them to find opportunities that align with their skills and preferences.
- **Application Process:** Job seekers can apply for jobs only if they have completed their profiles

to 100%. This requirement encourages job seekers to provide comprehensive information, improving their chances of landing suitable positions.

- **Notification System:** A robust notification system keeps both hirers and job seekers informed throughout the application process. Hirers receive notifications when a job seeker applies, allowing them to review and respond promptly.
- **Hirer Responses:** Hirers have the option to respond to job applications. Job seekers are instantly notified of these responses, facilitating real-time communication and follow-up.
- **Recommendation Algorithm:** JobHub incorporates a sophisticated recommendation algorithm that matches job seekers' skills with job postings requiring those skills. This personalized feature enhances the job-seeking experience, increasing the likelihood of successful matches.

II. INVESTIGATION

1. Plan of work:

A project with the features described is a multi-step process. Below is a plan of work for "JobHub" Django project with "job seeker" and "hirer" functionalities, including profile completion, job posting, job applications, notifications, filters, search bar, and a recommendation algorithm.

Step 1: Project Setup
- Set up of new Django project.
- Creating Django apps for "job seeker" and "hirer" functionalities.

Step 2: User Authentication
- Implement user authentication and registration for both "job seeker" and "hirer."

Step 3: User Profiles
- Creating user profile models for "job seeker" and "hirer" with fields for required information.
- Implement profile completion tracking for users (e.g., 0-100%).
- Creating forms and views for users to complete their profiles.

Step 4: Job Posting
- Design a job posting form for "hirers" to create job listings.
- Store job listings in the SQLite3 database.
- Implement validation to ensure only completed profiles can post jobs.

Step 5: Job Listings
- Creating views to display job listings to "job seekers."
- Implement filters, search bar, and sorting options for job listings.
- Display job recommendations based on skills.

Step 6: Job Applications
- Implement a form for "job seekers" to apply for jobs.
- Store job applications in the database, associating them with the corresponding job listing and user.
- Ensure that only users with completed profiles can apply.

Step 7: Notifications
- Implement a notification system for hirers and job seekers.
- Notify hirers when they receive job applications.
- Notify job seekers when hirers respond to applications.

Step 8: Job Application Responses
- Creating views for "hirers" to view job applications and respond to them.
- Update "job seekers" with responses via notifications.

Step 9: Testing and Debugging
- Thoroughly test all functionalities, including user profiles, job posting, job applications, and notifications. Debug and fix any issues that arise during testing.

III. METHODOLOGY

1. Functions for User Hirer:

- **Registration and Profile Creation:** The hirer begins by registering on the platform with their email or social media accounts.
- **Profile Completion Validation (100%):** The system ensures that the hirer's profile is 100% complete before allowing them to post job applications.
- **Job Posting:** Once the hirer's profile is verified, they can create and post job listings. This involves entering job details such as job title,

description, required skills, salary, location, and application deadline.
- **Storage in SQLite3 Database:** The job posting is stored in a SQLite3 database, making it accessible to job seekers.
- **Real-time Notifications:** The system instantly notifies the hirer when a job seeker applies for one of their posted jobs appears on website.
- **Application Review and Response:** The hirer reviews each job application received, evaluating the applicant's qualifications and fit for the role.
- **Profile Updates:** The hirer can update their profile information as needed, reflecting any changes in job listings or company details.
- **Logout:** At any point, hirers can log out of the platform to end their session securely.

2. Functions for User Job Seeker:

- **User Registration and Profile Creation:** Job seekers begin by registering on the platform using their email or social media accounts.
- **Profile Completion Validation (100%):** The system ensures that the job seeker's profile is 100% complete before they can apply for jobs.
- **Browsing Job Listings:** After profile completion, job seekers can browse job listings posted by hirers.
- **Search and Filter Features:** Job seekers can use search and filter features to find job listings that match their skills, preferences, and location.
- **Real-time Notifications:** Job seekers receive real-time notifications about new job listings that match their profile and preferences.
- **Applying for Jobs:** Job seekers can apply for jobs by submitting their applications, which include details about their qualifications and a cover letter.
- **Storage in SQLite3 Database:** Job applications are stored in a SQLite3 database, making them accessible to hirers for review.
- **Tracking Application Status:** Job seekers can track the status of their job applications, including whether they have been accepted or rejected by hirers.
- **Recommended Jobs:** The platform's recommendation algorithm suggests job listings that match the job seeker's skills and preferences, improving their chances of finding the right job.
- **Profile Updates:** Job seekers can update their profiles, including adding new skills or modifying their preferences.
- **Logout:** At any point, job seekers can log out of the platform to end their session securely.

3. Functions for Admin (Django Defaults Panel):

- **Accessing the Admin Panel:** The admin panel is accessible via a secure login page.
- **Authentication and Authorization:** Users with admin privileges are authenticated and authorized to access the admin panel.
- **Managing User Accounts:** Admins can view, create, update, and delete user accounts, including job seekers and hirers.
- **User Profile Management:** Admins have the ability to view and modify user profiles, helping users with any issues or updates required.
- **Job Listing Management:** Admins can manage job listings, including creating, editing, and deleting job postings on behalf of hirers.
- **Application Review:** Admins can review job applications submitted by job seekers, enabling them to take necessary actions such as approving or rejecting applications.
- **Data Management:** Admins have access to the SQLite3 database and can perform data management tasks, including backup, restoration, and data export.
- **User Activity Monitoring:** The admin panel provides logs and reports on user activities, helping admins track and audit actions taken on the platform.
- **Customization and Configuration:** Admins can customize the appearance and behavior of the admin panel, including adding custom views, models, and functionalities.
- **Security and User Roles:** Admins can manage user roles and permissions, ensuring that only authorized individuals can access certain features and data.
- **Troubleshooting and Support:** Admins can assist users, address issues, and provide support as needed, helping to maintain a smooth user experience.
- **Logs and Notifications:** The admin panel logs system events and provides notifications for important platform activities and updates.

IV. IMPLEMENTATION PLAN

The two-phase implementation plan for "JobHub," a Django-based job portal, starts with project setup and database design. Initially, we establish a development environment with Django and Python,

configuring essential settings and defining data models for job seekers, hirers, job listings, and notifications. User authentication and profile management are crucial components, allowing users to register and complete their profiles.

Hirers gain the ability to post job listings, with validation checks ensuring profile completeness. Job seekers benefit from powerful search and filtering features, while a recommendation algorithm enhances job matching based on skills. Real-time application submission, notification systems, and an intuitive UI with HTML, CSS, and AJAX are pivotal for user engagement. Scalability is a priority to handle increased user activity, positioning "JobHub" as a versatile and user-centric job portal.

The database schema will be straightforward, with tables for job listings, job seekers, hirers, and notifications. We'll implement user authentication and profile creation functionalities, enabling users to register and log in. Hirers will have the ability to post jobs, but only if their profiles are marked as complete.

Next, we'll build a basic job search feature for job seekers, allowing them to filter job listings based on a few key criteria. The project will focus on the application and notification system, demonstrating how job seekers can apply for jobs and hirers receive notifications when applications are submitted. While a full-scale recommendation algorithm can be intricate, we'll create a simplified version for basic skill matching between job seekers and job listings.

The user interface will be minimalistic, with HTML and CSS for layout and minimal styling, and JavaScript with AJAX for handling form submissions and displaying notifications. Testing will ensure the core features function correctly, and comprehensive documentation will provide instructions for using the application, providing a foundational web development project.

V. RESULT

The "JobHub" web development project, you will have a functional web application that connects job seekers and hirers in a simplified manner. Users will be able to register, log in, and create profiles. Hirers will have the ability to post job listings, with profile completeness checks in place to ensure valid postings. Job seekers can search for jobs based on basic criteria and apply for positions, triggering notifications to hirers.

While the recommendation algorithm will be basic, it will demonstrate the concept of skill matching between job seekers and job listings. The user interface will provide a straightforward layout with essential features for job posting, search, and application submission. It will also handle notifications through JavaScript and AJAX. Testing will ensure that core features work as intended, and documentation will guide users on how to navigate and utilize the application.

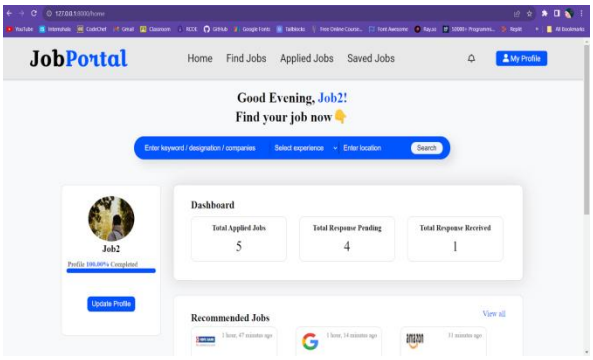

Fig 1. Website Dashboard for Job Seeker.

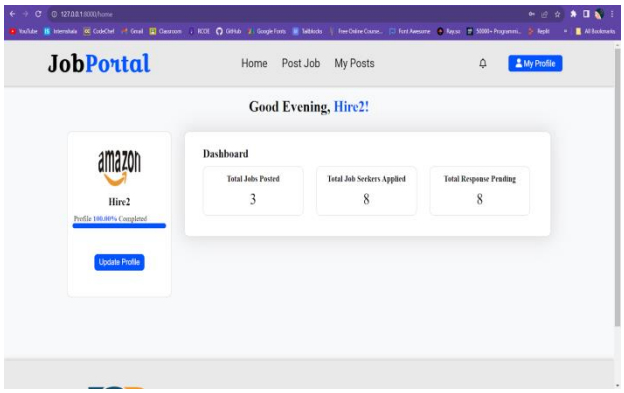

Fig 2. Website Dashboard for Hirer.

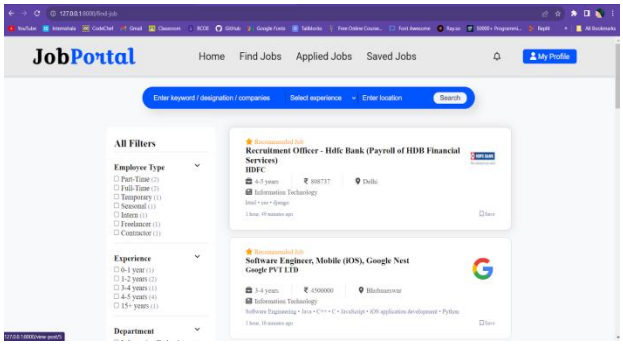

Fig 3. Website Find Jobs with search bar and filters.

VI. IMPLEMENTATION PLAN

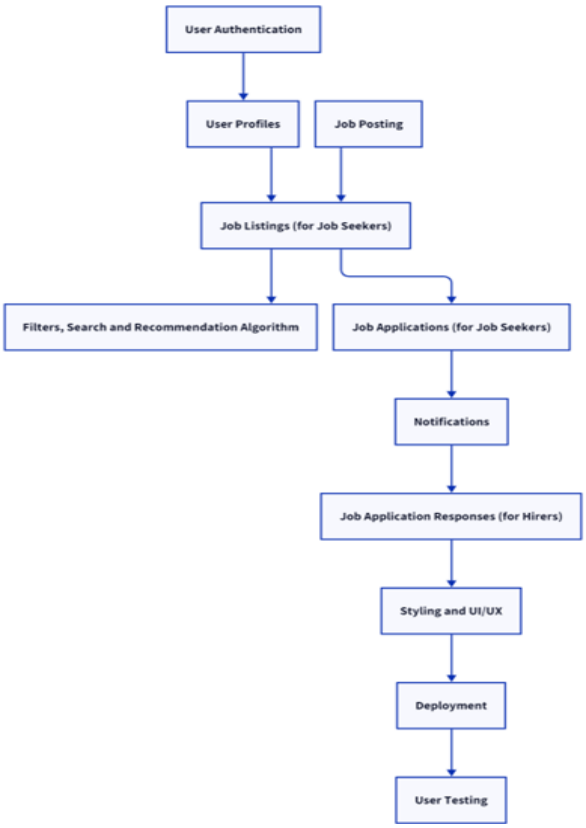

Fig 4. Implementation.

VII. CONCLUSION

The JobHub web development project represents a valuable learning experience for aspiring web developers. By completing this project, you have gained a foundational understanding of key web development technologies such as Django, HTML, CSS, JavaScript, and AJAX. You've successfully implemented essential features like user authentication, profile management, job posting, job search, and application submission, showcasing your ability to create a functional web application.

While this project is a simplified representation of a job portal, it provides a solid starting point for future endeavors in web development. The inclusion of a basic recommendation algorithm demonstrates the potential for expanding and enhancing the platform with more sophisticated features in the future. Overall, this project underscores the importance of user-centric design, effective testing, and comprehensive documentation in the development process, setting a strong foundation for your future web development projects and aspirations.

REFERENCES

[1] Django for Beginners by William S. Vincent.
[2] Two Scoops of Django by Daniel Roy Greenfeld and Audrey Roy Greenfeld.
[3] Mahek Katariya, Nigam Patel, Harsh Shah, Birla Vishvakarma Mahavidyalaya Engineering College, April 2020.
[4] A Sandhya, Murali Ponaganti, Telangana Online Job Portal Application, July 2022