TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES - MANILA

Bachelor of Science in Information Technology

College of Computer Studies

S.Y. 2025 - 2026

FINAL PROJECT

"EasyHire: An Efficient Job Portal System Built with Django"

Web System and Technologies 2

CIT 511 - IT32S2

Submitted by:

Ferrancullo, Rica

Gabuya, Andreah

Mangahas, King Angelo B.

Ravina, Yves

Submitted to:

Mr. Francis Carabuena

Date:

TABLE OF CONTENTS

Title Proposal: “EasyHire: An Efficient Job Portal System Built with Django”
Title Reference: Jobisite (https://www.jobisite.com/)

I. Introduction

Background of the Study

The integration of social media with job portals has significantly transformed recruitment strategies. Platforms like LinkedIn, Facebook, and Twitter have enabled employers to tap into a vast talent pool, facilitating direct engagement with potential candidates. This approach has proven effective in attracting passive candidates and strengthening employer branding. A report by Glassdoor revealed that 79% of job seekers utilize social media in their job search, underscoring its growing importance in modern recruitment.

Perceptions of Job Seekers Towards Digital Job Portals

Understanding job seekers' perceptions of digital job portals is crucial for enhancing user experience and effectiveness. A qualitative study by Sarath and Sandhya (2022) examined factors such as:

- Trustworthiness
- Website aesthetics
- Service quality
- Cost-effectiveness
- Content quality

The study found that while job portals are generally perceived positively, areas like cost-effectiveness and content quality require improvement to meet user expectations.

Impact on Employee Performance

The influence of online job portals extends beyond recruitment, impacting employee performance as well. Ahmed (2023) reviewed the effect of these platforms on employee performance, highlighting benefits such as:

- Streamlined hiring processes
- Better job-role alignment
- Increased job satisfaction

The study concluded that online job portals contribute positively to organizational efficiency and employee productivity.

Challenges and Limitations

Despite their advantages, online job portals face challenges, including the prevalence of "ghost jobs"—positions advertised but never filled. Reports indicate that up to 22% of job listings fall into this category, leading to frustration among job seekers and emphasizing the need for greater transparency in job postings.

Justification of the Study

In the modern digital landscape, an efficient, secure, and user-friendly website is crucial for ensuring a positive user experience. The identified challenges, such as poor navigation, weak authentication mechanisms, inefficient search functionality, and lack of personalization, hinder the effectiveness of the website. Addressing these issues through a structured approach will significantly improve usability, security, and performance.

Implementing Django, social authentication, and a recommender system will provide practical solutions to these challenges.

II. Statement of the Problem

In today’s digital world, websites are the main platform for organizations to connect with users. However, several challenges affect the website’s functionality, usability, and security. This study aims to identify and address these issues to improve user experience, accessibility, and overall performance.

a. General Problem

The website is experiencing several usability, security, and performance issues that negatively impact user experience and engagement. These challenges include difficulties in navigation, slow loading times, poor mobile responsiveness, ineffective search functionality, and security concerns. Without addressing these problems, users may find it frustrating to interact with the platform, potentially reducing its effectiveness and accessibility.

With these problems our system should develop:

- **Improved Navigation System** which will enhance the website’s structure and layout to ensure intuitive and seamless navigation.

- **Optimized Performance** which will reduce loading times by implementing efficient coding practices, caching, and content delivery networks.

- **Responsive Design** which will ensure the website is fully optimized for various screen sizes and devices to improve mobile accessibility.

- **Enhanced Search Functionality** which will develop an advanced search system with filters and predictive text to improve content discovery.

- **Security Enhancements** which will implement strong authentication, data encryption, and protection against common cyber threats.

b. Specific Problems

Social Problems

- **User Trust and Privacy Concerns**: Many people are hesitant to share their personal information online because they're worried about their data being stolen or misused. This makes them trust the website less and reduces their willingness to use it.

- **Digital Literacy Gaps:** Some users may not have the technical skills or knowledge to fully understand or use more advanced features of the website. This can lead to confusion, frustration, or them leaving the site.

- **Exclusion of Certain Demographics**: If the website isn't designed to be accessible, users with disabilities (like visual or hearing impairments) or people from different backgrounds may struggle to use it, leading to frustration and lower user satisfaction.

Technical Problems

- **Lack of Optimization for Diverse Devices:** The website may not work well on different devices, especially mobile phones. If it's hard to use on smaller screens, users may leave the site, reducing its effectiveness.

- **Scalability Issues:** The website may not be able to handle large numbers of visitors at once. This can cause the site to slow down or crash, which makes it harder for people to use during busy times.

- **Poor Database Management:** If the website's data storage isn't organized well, it can take longer to retrieve information. This delays loading times and makes the site feel slow.

- **Insecure API Integrations:** The website might rely on third-party services (APIs) that are not secure enough. If these services are compromised, users' data could be at risk, leading to potential security breaches.

III. Scope and Limitations

a. Scope

The **EasyHire: An Efficient Job Portal System Built with Django** project aims to develop a user-friendly web platform that addresses several challenges commonly faced by job seekers and employers. The system will be built using the:

- **Django framework**, which will provide a strong, secure, and organized backend for managing user data, job listings, and other necessary features. One of the core features of the system will be **social media authentication**, which allows users to sign up or log in easily using their existing accounts from platforms like Facebook, Google, or LinkedIn. This will improve the login process, making it both faster and more secure.

Another major feature of the platform will be an:

- **Advanced search and filtering system**. Users will be able to filter job listings based on different criteria such as job title, location, salary, and experience level. This will solve the problem of ineffective search functionality and make it easier for users to find relevant job opportunities. To improve the user experience even further, the platform will have a clean, **intuitive design** with an easy-to-navigate structure, addressing the problem of cluttered layouts and difficult navigation. The system will also be **optimized for mobile devices**, ensuring that users can access and use the platform smoothly on smartphones and tablets, solving the issue of poor mobile responsiveness.

To ensure the security of users' personal data and accounts, the project will incorporate:

- **Multi-factor authentication (MFA)** to make the login process more secure. This will help solve problems related to weak authentication and improve overall security. Additionally, the platform will use **encryption and data protection** methods to keep user information safe from potential threats. Performance is another key focus of the project. The website will be designed to load quickly by using **optimized coding practices**, caching, and content delivery networks (CDNs), addressing the problem of slow loading times and improving the platform's overall performance.

b. Limitations

While the project aims to cover these essential features, there are some limitations. One major constraint is the:

- **Time available** for development. Due to the limited project timeline, certain advanced features might be deprioritized or left out for the initial launch. The project will focus on building a functional system, with additional features potentially being added in the future.
- **Data availability**. Since the platform will rely on real-world job listings and user data to make job recommendations, a lack of access to a large dataset during development might affect the accuracy of recommendations at the start.

The platform will be built to handle a moderate amount of users, but it might face **scalability challenges** if the website experiences very high traffic. In this case, additional resources may be needed to ensure smooth performance. Another limitation is the:

- **Dependence on third-party services** like social media logins. If these services experience issues or change their API features, it could affect the user experience.
- Lastly, the platform may not support **real-time job updates** in its initial version. Although job listings will be regularly updated, real-time updates would require additional technical resources and might be included in a future version.

In summary, while the **EasyHire** project will address key issues such as poor navigation, weak security, and slow performance, it also has some constraints related to time, data availability, and scalability. The system will be developed with these limitations in mind, ensuring a functional and secure platform for users.

IV. Significance of the Study

The EasyHire job portal system will benefit job seekers, employers, and businesses by improving their overall experience. Job seekers will enjoy a simpler and faster way to find jobs, with advanced search options, personalized recommendations, and easy login through social media which our system EasyHire will emphasize. Within this employers will have a more efficient way to post jobs and find suitable candidates, saving time and effort.

The project will enhance security with features like multi-factor authentication and improve user engagement through better navigation and mobile optimization, leading to higher user satisfaction. Finally, businesses will be able to recruit more effectively and reach a larger pool of potential candidates, improving their hiring process.

Overall, the project will make the platform more user-friendly and efficient, benefiting all parties involved by making the job search and hiring process smoother and more secure.

V. Methodology

The development of the **EasyHire** job portal system will involve the use of several technologies and methodologies to ensure an efficient, secure, and user-friendly platform. Below is an overview of the technology stack and development process:

1. **Technology Stack**

   - The project will be built using **Django**, a robust and secure Python-based framework. For the database, either **PostgreSQL** or **MySQL** will be used to store and manage user and job-related data efficiently.

   - To enhance the front-end, **JavaScript** will be integrated for dynamic features and interactive elements on the website.

   - The backend will be further enhanced by **Django Rest Framework** to create RESTful APIs for seamless communication between the server and the client-side of the application.

2. **Social Authentication**: To make the login process smoother, the system will implement **social authentication** using **Django Allauth** or **OAuth**. This will allow users to sign in using their existing accounts from popular platforms like **Google** or **Facebook**, making the registration and login process quicker and more secure.

3. **Recommender System**: The platform will include a **recommender system** to personalize job recommendations for users. This system will use **machine learning models** such as **collaborative filtering** and **content-based filtering** to suggest jobs based on users' previous activities, preferences, and similar user behavior. This feature will improve job matching and enhance the user experience by helping users find relevant job listings more easily.

4. **Development Process**: The development process will follow an **Agile methodology**, utilizing **Scrum** or other iterative approaches. This will allow for continuous feedback, regular updates, and a flexible development cycle. The project will be divided into manageable sprints to ensure steady progress and to address any issues or improvements as they arise. Regular testing and integration will be conducted to ensure the platform meets the required standards.

Implementing this methodology will help in delivering a well-structured, user-friendly, and functional job portal that can scale with increasing user demands and provide a positive experience for both job seekers and employers.

VI. Expected Output

- A Django-based platform with social authentication and a recommender system
- Enhanced user engagement and personalization

VII. Review of Related Literature

- Cite relevant studies on authentication, recommender systems, and Django web development

VIII. Project Timeline

- Provide a Gantt chart or timeline outlining key milestones

Project Timeline

Day 1

On March 29, 2025, we began by addressing security concerns in the Django project. We identified that the project needed proper OAuth configurations for social authentication. We implemented Google OAuth credentials first, configuring the client ID and secret key in the settings. When testing the Google authentication, we encountered HTTPS-related issues with the development server, which led us to modify security settings to allow HTTP for local development.

Moving to Facebook authentication, we added the Facebook OAuth credentials and configured the necessary settings. This process revealed issues with browser caching and HSTS settings, which we resolved by clearing browser security policies. We also fixed template-related problems in the login.html file to properly display social login buttons.

Day 2

On March 30, 2025, we focused on resolving the MultipleObjectsReturned error that was occurring due to duplicate social applications in the database. We properly configured all three social providers (Google, Facebook, and GitHub) in the Django admin panel, ensuring each had the correct credentials and site configurations. We also updated deprecated settings in the project to use the newer django-allauth configurations.

Throughout these two days, we maintained focus on security while making the authentication system work in the development environment. The project now has a functioning social authentication system with Google, Facebook, and GitHub integration, though we noted that additional features like user management and core job portal functionality still need to be implemented.

The current state of the project provides a solid foundation for social authentication, ready for the next phase of development to build the actual job portal features.

IX. References

- List academic papers, books, and resources related to Django and recommendation systems