**HIGHTAIL**

# A Project Report

Submitted in partial fulfilment of the

Requirements for the award of the

Degree of

**BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)**

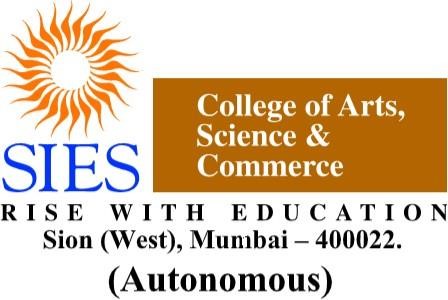
**By**

**Prajin Josemon Puthiri**

**TIT2425045**

**Under the esteemed guidance of**

# Mrs. Anita Gupta



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**SIES COLLEGE OF ARTS, SCIENCE & COMMERCE**

**(AUTONOMOUS)**

**SION (W), MUMBAI, 400022**

**MAHARASHTRA**

**2024**

### PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

Roll No**:** TIT2425045

|  |  |
| --- | --- |
| 1. Name of the Student | Prajin Josemon Puthiri |
| 1. Title of the Project | HIGHTAIL |
| 1. Name of the Guide | Mrs. Anita Gupta |
| 1. Teaching experience of the Guide | 12 years |
| 1. Is this your first submission? | Yes No |

|  |  |
| --- | --- |
| **Prajin Josemon Puthiri** | **Mrs. Anita Gupta** |
| Signature of the Student | Signature of the Guide |
| Date: **06/07/2024** | Date: **06/07/2024** |

**Mrs. Sudha. B**

Signature of the Coordinator

Date: **06/07/2024**

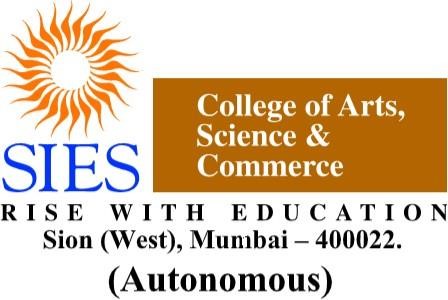
## SIES COLLEGE OF ARTS, SCIENCE & COMMERCE (AUTONOMOUS)

**(Affiliated to University of Mumbai)**

**SION (W), MUMBAI-400022**

**DEPARTMENT OF INFORMATION**

**TECHNOLOGY**



**CERTIFICATE**

This is to certify that the project entitled, **"VITTAVIDYA"**, is bonafide work of **Mr. Ayush Bhajman Patra** bearing Seat No: **TIT2425043 s**ubmitted in partial fulfilment of the requirements for the award of degree of **BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY.**

Mrs. Anita Gupta Mrs. Sudha.B

**Internal Guide Coordinator**

**Date: 06/07/2024 External Examiner**

**Abstract**

Hightail is a critical platform for professionals seeking to expand their networks and advance their careers. Users create inclusive profiles showcasing their professional experience, skills, and accomplishments, making it a digital resume of sorts. Beyond personal profiles, Hightail serves as a hub for connecting with colleagues, recruiters, and industry peers, facilitating opportunities for job searching and recruitment. It also promote knowledge sharing through articles, posts, and discussions, allowing individuals to stay informed about industry trends and developments. Overall, Hightail plays a crucial role in modern professional life by providing tools for networking, career growth, and continuous learning.

**ACKNOWLEDGEMENT**

I extend my heartfelt appreciation to my parents and well-wishers whose unwavering support was instrumental in the successful completion of this project.

I would like to convey our sincere gratitude to our mentor, **Ms. Anita Gupta**, who served as our guiding light throughout the entire journey of conceptualizing and executing this project. Her expertise, encouragement, and invaluable insights played a pivotal role in its realization.

I also wish to acknowledge the unwavering support of the Head of the IT Department, **Mrs. Sudha** **Bhagavatheeswaran**. Her leadership and encouragement bolstered our determination and were integral to the successful completion of this endeavor.

**DECLARATION**

I hereby declare that the project entitled, “**HIGHTAIL**” done at **SIES COLLEGE OF ARTS, SCIENCE AND COMMERCE (EMPOWERED AUTONOMOUS)**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

**Prajin Josemon Puthiri**

**Name and Signature of the Student**

**Table of Contents**

|  |  |
| --- | --- |
| 1. **Chapter: Introduction**    1. Background    2. Objectives    3. Purpose, Scope, Applicability   1.3.1. Purpose  1.3.2. Scope  1.3.3. Applicability | **1**  1  1  2  2  2  3 |
| 1. **Chapter: Survey of Technologies**    1. Front-End Development   2.1.1. Front-End Development Languages   * + 1. Front-End Framework   2.2. Back-End Development  2.2.1. Server-side scripting Languages  2.2.2. Back-End Framework  2.2.3. Database Management Languages  2.3. Technologies used in this project | **4**  4  4  8  9  9  12  13  16 |
| 1. **Chapter: Requirements and Analysis**    1. Problem Definition    2. Requirements Specification    3. Planning and Scheduling       1. Planning       2. Scheduling    4. Software and Hardware Requirement       1. Software Requirements       2. Hardware Requirements    5. Conceptual Models       1. Data Flow Diagram       2. Entity-Relationship Diagram       3. Activity Diagram       4. Sequence Diagram       5. UseCase Diagram | **17**  17  17  19  19  20  22  22  22  23  23  23  26  27  28 |
| 1. **Chapter: System Design**    1. Basic Modules    2. Schema Design    3. User Interface Designs    4. Security Issues   References | **29**  29  31  34  37  38 |

**List of Tables**

|  |  |
| --- | --- |
| Table 3.1. Critical Activity | 20 |
| Table 4.2.1. User Module  Table 4.2.2. Admin Module  Table 4.2.3. User Post Module  Table 4.2.4. Admin Post Module  Table 4.2.5. Schema Module | 27 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| Figure 3.1. Gnatt Chart  Figure 3.2. Activity  Figure 3.3. Critical Activity  Figure 3.4. Level 0 Data Flow Diagram  Figure 3.5. Level 1 Data Flow Diagram  Figure 3.6. Entity-Relationship Diagram  Figure 3.7. Activity Diagram  Figure 3.8. Sequence Diagram  Figure 3.9. Use Case Diagram | 19  20  20  21  22  23  24  25  26 | |
| Figure 4.1. Loading Page  Figure 4.2. Home Page  Figure 4.3. Login Page  Figure 4.4. Sign up Page  Figure 4.5. User Sign up Page  Figure 4.6. Admin Sign up Page  Figure 4.7. User Profile page  Figure 4.8. Admin Profile page  Figure 4.9. Post Page | | 27  28  28  29  29  30  30  31  31 | |

**Chapter 1: Introduction**

* 1. **Background**

Hightail has made significant contributions in several areas related to professional networking, career development, and business growth. And after a pandamic yrs of covid-19 many freshers can not get jobs. So the hightail will help them to get experience and opportunities to work with various companies. hightail has transformed the job search process by providing a centralized platform for job seekers to showcase their skills, experience, and accomplishments. Recruiters and hiring managers use hightail extensively to find and evaluate candidates based on their profiles, recommendations, and endorsements.

**1.2. Objectives**

Hightail’s primary objectives are centered on connecting professionals worldwide, facilitating career growth, and fostering business development. As a leading professional networking platform, it enables individuals to build and maintain their professional networks, creating opportunities for collaboration, mentorship, and career advancement. It serves as a core for job seekers and recruiters alike, providing tools for job searching, recruitment, and candidate evaluation based on detailed profiles highlighting skills and achievements. Hightail also supports personal branding by allowing users to showcase their expertise and accomplishments, thereby increase reputation and visibility within their industries. For businesses, Hightail offers tools to establish and expand their presence, connect with potential clients and partners, and engage in targeted marketing efforts. promoting a supportive and inclusive professional environment globally.

* 1. **Purpose, Scope, Applicability**
     1. **Purpose**

Hightail serves a pivotal purpose as the premier online platform for professional networking and career development. At its core, Hightail enables individuals to connect with peers, mentors, and industry leaders globally, fostering opportunities for collaboration, knowledge sharing, and career advancement. It functions as a digital resume, allowing users to showcase their skills, experience, and achievements, thereby enhancing their professional credibility and visibility. Beyond networking, Hightail supports businesses by providing tools for establishing a corporate presence, engaging with clients and partners, and recruiting talent. It also serves as a hub for continuous learning through Hightail Learning, offering a diverse range of courses to develop and enhance professional skills. Moreover, Hightail facilitates insights into industry trends, job market dynamics, and business intelligence, leveraging its extensive user data. Overall, Hightail's purpose is to empower professionals and businesses alike, facilitating connections, fostering growth, and driving innovation in the global professional community.

* + 1. **Scope**

VITTAVIDYA provides learners with access to a comprehensive library of e-books and instructional videos created by experienced professionals in the field of forex trading. These e-books cover a wide range of topics, of the basics of forex trading and market analysis techniques. The instructional videos, often featuring industry experts, offer detailed explanations and visual demonstrations of key concepts, making complex topics more accessible and easier to understand. VITTAVIDYA online learning application typically offers video lectures that learner can watch at their own pace. VITTAVIDYA assists the learner to learn new concepts or review topics that they have already learned. The platform regularly updates its content to reflect the latest market trends and strategies, ensuring that learners are always equipped with current knowledge. That acts as a great platform where learner can get support and feedback of their learning. VITTAVIDYA offers opportunity to the learner to explore more about the forex. VITTAVIDYA helps the Learner to understand the concept clearly. VITTAVIDYA helps the learner to get the practical knowledge and the daily usage of the concept by using live graphs. By combining theoretical knowledge with practical application, VITTAVIDYA ensures that learners are well-prepared to navigate the complexities of forex trading. Thus, VITTAVIDYA enriches overall development of learner.

* + 1. **Applicability**

Hightail serves as a inclusive platform that facilitates professional networking, career development, and business growth on a global scale. It enables individuals to connect, build relationships, and showcase their skills and accomplishments to potential employers and collaborators. For businesses, Hightail offers tools for establishing a strong online presence, engaging with target audiences, and generating leads. The platform also supports continuous learning through Hightail Learning, providing a wide range of courses to enhance professional skills. Hightail's insights into industry trends and market dynamics empower organizations to make strategic decisions, while its commitment to diversity and inclusion fosters an environment of equity and opportunity for all professionals. Overall, Hightail plays a crucial role in connecting professionals, advancing careers, and driving innovation in the global workforce.

**Chapter 2: Survey of Technologies**

A web application is a software application that is accessed and used through a web browser. It is a client-side and server-side software application in which the client runs or requests in a web browser. It is hosted on a web server and can be accessed by users from anywhere with an internet connection. Web applications are popular because they don't require any installation on the user's device. A web application typically consists of two main components, the front end and the back end. The front end of a web application refers to the user interface that the user interacts with directly. It is the visible part of the application and is responsible for presenting data. The back end of a web application is the server-side of the application that runs on the web server. It is responsible for processing requests from the front end, performing operations, interacting with databases, and sending responses back to the client (the user's web browser).

**2.1. Front-End Development**

Front-end development, also known as client-side development, is the process of building the user interface and user experience (UI/UX) of a web application or website. Front-end development is critical for creating an attractive, user-friendly, and performant website. Frontend languages and frameworks contribute to the overall front-end development. A front-end developer creates the overall design, in addition to debugging and using the static code analysis.

**2.1.1. Front-End Development Languages**

Front-end development languages refer to the programming languages used by front-end developers to create the user interface (UI) and user experience (UX) of a web application. These languages are responsible for structuring the content, applying visual styles, and adding interactivity to the web pages that users directly interact with through their web browsers.

**2.1.1.1. Hyper Text Markup Language (HTML)**

HTML (Hypertext Markup Language), being the foundational language of the web, played a crucial role in shaping the development of other front-end languages. HTML has indeed played a fundamental role in shaping the development of other front-end languages and technologies. As the foundational language of the web, HTML provided the groundwork for how web content is structured and displayed. Its importance in shaping the front-end development landscape cannot be overstated. HTML allows developers to organize and present information on web pages, forming the backbone of every website. The latest version, HTML5, introduced new features like support for multimedia, semantic elements, and improved form elements, making it even more powerful for modern web development. HTML works alongside CSS for styling and JavaScript for interactivity, enabling the creation of dynamic and visually appealing web applications.

**Advantages:**

* Ease of Use: HTML uses a simple and straightforward syntax of tags and elements, making it easy to learn and use, even for beginners.
* Platform Independence: HTML is a platform-independent language, meaning web pages created with HTML can be viewed and accessed on any device with a web browser.
* Wide Adoption and Standardization: HTML is the standard markup language for the web, which ensures widespread adoption and support across web browsers and devices.
* Scalability: HTML is suitable for both simple static web pages and complex web applications.

**Disadvantages:**

* Lack of Interactivity: HTML on its own is a static language and lacks built-in mechanisms for creating dynamic and interactive elements.
* Limited Styling Capabilities: HTML's styling capabilities are limited compared to CSS (Cascading Style Sheets).
* Complexity for Large Projects: For large-scale web applications, HTML alone may not be sufficient to manage the complexity of the project.

**2.1.1.2 Cascading Style Sheets (CSS)**

CSS stands for "Cascading Style Sheets." It is a programming language used to style the appearance and layout of web pages written in HTML (Hyper Text Markup Language) or XML (extensible Markup Language). CSS provides a way to control the presentation and formatting of web content, including elements such as text, images, and other HTML elements. The primary purpose of CSS is to separate the content (HTML) from the presentation (styling). By doing so, web developers can maintain a clear distinction between the structure of the page and its visual representation. This separation allows for greater flexibility and easier maintenance of web pages.

**Advantages:**

* Consistency and Efficiency: CSS enables consistent styling across an entire website. By defining styles once and applying them to multiple elements, developers can efficiently manage the presentation of the website and avoid repetitive code.
* Page Load Speed: CSS files are lightweight and can be cached, leading to faster loading times for subsequent visits to a website. This is crucial for improving user experience and search engine rankings.
* Maintainable Codebase: With CSS, developers can keep their code modular and organized using classes and IDs, making it easier to manage and update styles in large-scale projects.

**Disadvantages:**

* Complexity: CSS can become complex, especially when styling large and intricate web pages. As the project grows, managing and organizing CSS code can become challenging, leading to potential maintenance difficulties and increased development time.
* Lack of Variables: CSS lacks native support for variables until CSS3 introduced custom properties (variables). Prior to that, developers resorted to workarounds like using preprocessors like Sass or Less to handle variables.
* Lack of Encapsulation: CSS rules are global by default, and it may be difficult to encapsulate styles to avoid unintended effects on other elements. This can be mitigated using CSS methodologies like BEM (Block Element Modifier) or CSS-in-JS solutions.

**2.1.1.3. JavaScript**

JavaScript is a versatile and widely used programming language primarily used to add interactivity, behavior, and dynamic features to websites. It is often referred to as the "language of the web" because it runs directly in web browsers, allowing developers to create interactive and responsive user interfaces. JavaScript is an essential component of modern web development and works in tandem with HTML and CSS to build rich web applications.

**Advantages:**

* Client-side Interactivity: JavaScript allows developers to add interactive elements and behaviors to web pages. It can handle user actions, respond to events, and update the content dynamically without requiring a full-page refresh. This creates a more engaging and responsive user experience.
* Platform Independence: JavaScript runs on the client-side, within web browsers, making it platform-independent.
* Rich Ecosystem: JavaScript has a vast ecosystem of libraries, frameworks, and plugins that make web development more efficient.
* Single Codebase for Both Front-end and Back-end: With Node.js, JavaScript can be used on the server-side as well, allowing developers to use a single programming language for both front-end and back-end development, streamlining the development process.

**Disadvantages:**

* Security Concerns: Because JavaScript runs on the client-side, it can be vulnerable to security threats such as cross-site scripting (XSS) attacks. Malicious users may exploit vulnerabilities in JavaScript code to execute harmful actions on users' browsers.
* Lack of Native Multithreading: JavaScript is single-threaded, meaning it cannot natively take advantage of multiple processor cores.
* Browser Compatibility: Different browsers may interpret JavaScript code slightly differently, leading to cross-browser compatibility issues.

**2.1.2. Front-End Frameworks**

Front-end frameworks are pre-built sets of tools, libraries, and guidelines designed to facilitate the process of building user interfaces (UI) for web applications. They provide a structured and standardized approach to front-end development, offering reusable components, styling, and functionality that help developers create visually appealing, responsive, and interactive websites and web applications more efficiently.

**2.1.2.1. React**

React, also known as React.js or ReactJS, is an open-source JavaScript library developed and maintained by Facebook. It is a popular front-end library for building user interfaces and is widely used in modern web development. React was first released in 2013 and has since gained immense popularity due to its efficiency, performance, and reusability.

**Advantages:**

* Component-Based Architecture: React follows a component-based approach, which promotes code reusability and modularity.
* It ensures fast rendering.
* It guarantees stable code.

**Disadvantages:**

* Boilerplate Code: React applications may require additional setup and boilerplate code to configure build tools, state management libraries (if used), and other necessary dependencies.
* SEO Challenges: React applications are typically rendered on the client-side, which can cause challenges with search engine optimization (SEO).
* Component Hierarchy Complexity: As applications grow, the component hierarchy in React can become complex, leading to difficulties in managing state and data flow between components.
* Large Bundle Sizes: Including all the necessary dependencies and libraries can result in large bundle sizes, affecting the initial loading time of the application.

**2.2. Back-End Development**

Backend development refers to the process of building and maintaining the server-side of a web application or software. It is concerned with managing the logic, databases, server, and all the aspects that enable the front-end (client-side) to function properly. In a client-server architecture, the backend is responsible for processing data, handling requests from the frontend, and ensuring the smooth functioning of the application.

**2.2.1 Server-side scripting languages**

Server-side scripting languages are programming languages that are executed on the server side of a web application. These languages are used to process and handle server-side logic, data, and interactions, and they generate dynamic content that is then sent to the client-side (typically a web browser). Server-side scripting is an essential part of web development, enabling the creation of dynamic and interactive web applications.

**2.2.1.1. NodeJS**

Node.js is an open-source, cross-platform, back-end, JavaScript runtime environment that executes JavaScript code outside a web browser. Node is useful for developing applications that require a persistent connection from the browser to the server and is often used for real time applications. Node.js represents a "JavaScript everywhere" paradigm, unifying web application development around a single programming language, rather than different languages for server- and client-side scripts.

**Advantages:**

* JavaScript: Node.js uses JavaScript, which is a popular programming language that is known for its simplicity and ease of use.
* Performance: Node.js is very performant. It uses a single-threaded event loop, which makes it very efficient at handling asynchronous requests.
* Scalability: Node.js is very scalable. It can be easily scaled horizontally by adding more nodes to the cluster.

**Disadvantages:**

* Unstable API: Node.js's API is constantly changing, which can make it difficult to keep up with the latest changes. This can be a problem for developers who are using older versions of Node.js.
* Lack of library support: Node.js has a large library ecosystem, but there are some areas where the library support is lacking. This can make it difficult to find libraries for specific tasks.

**2.2.1.2. PHP**

PHP's role in the development of web applications is akin to the foundation upon which a sturdy structure is built. It seamlessly integrates with front-end technologies, enhances server-side dynamics, and offers the versatility needed to create robust, secure, and high-performance web solutions. Its continued evolution and vibrant community make it an enduring force in the world of web development.

**Advantages:**

* Open Source: PHP is open-source software, which means it's free to use, modify, and distribute.
* Versatility: PHP is versatile and can be used for a range of web-related tasks, including server-side scripting, command-line scripting, and even desktop applications. It's not limited to web development.
* Database Integration: PHP seamlessly integrates with popular databases like MySQL, PostgreSQL, SQLite, and others. This feature makes it an excellent choice for building database-driven web applications.
* Security: PHP offers features and practices for building secure web applications, including input validation, encryption libraries, and authentication mechanisms.

**Disadvantages:**

* Weak Typing: PHP is a loosely typed language, meaning variables are not explicitly defined with data types.
* Inconsistent Naming Conventions: PHP's standard library and functions have inconsistent naming conventions, which can be confusing for developers.
* Limited Multithreading: PHP traditionally follows a shared-nothing architecture, which means each request is handled by a separate PHP process.

**2.2.1.3 C#**

C# (pronounced “C-sharp”) is a modern, object-oriented programming language developed by Microsoft as part of its .NET initiative. It is designed for building a wide range of applications, from web and desktop applications to mobile and gaming.

**Advantages:**

* C# is a true object-oriented language, promoting concepts like encapsulation, inheritance, and polymorphism, which lead to more modular, flexible, and maintainable code.
* C# includes modern language features like async/await, LINQ, and dynamic programming, making it suitable for various applications, from web services to game development.
* The .NET framework and .NET Core provide extensive libraries, which simplify many tasks like file handling, data access, and web services.
* Being backed by Microsoft, C# has a robust community, extensive documentation, and strong support, ensuring that developers have access to a wealth of resources.
* With the introduction of .NET Core, C# can now be used to build cross-platform applications that run on Windows, macOS, and Linux.
* C# integrates seamlessly with other Microsoft products and services, such as Azure for cloud computing and Visual Studio for development.

**Disadvantages:**

* Historically, C# and the .NET framework were closely tied to Windows, though .NET Core has alleviated much of this concern by enabling cross-platform development.
* Although C# has garbage collection, developers still need to be mindful of memory management, particularly with unmanaged resources.
* While C# performance is generally good, it may not match the performance of languages like C or C++ for low-level system programming.
* For beginners, C# can have a steep learning curve, especially with its vast ecosystem and the complexity of the .NET framework.
* The .NET framework is extensive and can be overwhelming for new developers to fully grasp.

**2.2.2. Backend Frameworks**

Backend frameworks are software frameworks that provide a structured and organized way to build server-side applications and manage backend functionalities. These frameworks facilitate the development process by offering pre-built components, libraries, and tools that handle common tasks, such as routing, database interactions, authentication, and session management. Backend frameworks help developers focus on the application's business logic rather than reinventing the wheel for routine functionalities.

**2.2.2.1. ExpressJS**

Express.js is a minimalist and flexible web framework for Node.js, a runtime environment for server-side JavaScript. It simplifies building web applications and APIs using Node.js by providing a straightforward and unopinionated approach to web development. It is lightweight and helps to organize web applications on the server-side into a more organized MVC architecture.

**Advantages:**

* Middleware Support: Express.js is built around the concept of middleware, which allows developers to add extra functionality to the application's request-response cycle.
* Routing: Express.js provides a powerful and flexible routing system, allowing developers to define routes for different HTTP methods and URLs.
* Easy to integrate with different template engines like Jade, Vash, EJS.

**Disadvantages:**

* Async Error Handling Complexity: Handling errors in asynchronous code can be more complex in Express.js.
* Maintainability with Middleware: As applications grow, the use of multiple middleware can lead to complex request/response chains that can be difficult to maintain and debug.

**2.2.2.2 ASP.NET**

The original ASP.NET is a mature framework for building robust web applications on Windows. ASP.NET is an open-source web framework, created by Microsoft, for building modern web apps and services with. NET. ASP.NET is cross platform and runs on Linux, Windows, macOS, and Docker.

**Advantages:**

* Mature and stable.
* Extensive libraries and tools.
* Strong integration with Windows Server and IIS.

**Disadvantages:**

* Windows-only, limiting cross-platform capabilities.
* Heavier and less flexible compared to ASP.NET Core.

**2.2.3. Database Management Languages**

Database management languages are specialized languages used to interact with and manage databases. These languages provide a standardized way to perform various operations on databases, such as querying, updating, inserting, and deleting data. They act as intermediaries between applications and the underlying database systems, allowing developers and administrators to interact with databases using structured and efficient commands.

**2.2.3.1. MySQL**

MySQL is an open-source relational database management system (RDBMS) that is widely used for web-based applications and other data-driven software. MySQL is a freely available database system. However, there are several paid editions also available with which you can use advanced functionality. MySQL is easy to use as compared to other database software such as Microsoft SQL Server and Oracle database etc. It can be used with any programming language but is largely used with PHP, Python and Node.js. MySQL can run on multiple platforms such as Linux, Windows, Unix, and an information schema to define and manage your metadata. It is a really flexible, scalable, fast, and reliable solution.

**Advantages:**

* Relational Database Model: MySQL follows the relational database model, allowing you to organize data into structured tables with well-defined relationships between them. This makes it easier to maintain data integrity and perform complex queries efficiently.
* High Performance and Scalability: MySQL is designed for high performance and can handle large datasets and high traffic loads. With proper indexing and optimization, MySQL can deliver fast response times even for complex queries.
* Cross-Platform Compatibility: It is compatible with most of the operating systems including Windows, Linux, NetWare, Novell and other variations of UNIX.

**Disadvantages:**

* MySQL is not very efficient in handling very large databases.
* MySQL is prone to data corruption as it inefficient in handling transactions.
* MySQL versions less than 5.0 do not support COMMIT, stored procedure and ROLE.

**2.2.3.2. MongoDB**

MongoDB is a popular, open-source NoSQL database designed for handling large volumes of unstructured or semi-structured data. Unlike traditional relational databases that use tables and rows, MongoDB stores data in flexible, JSON-like documents called BSON (Binary JSON). This schema-less design allows for rapid development and easy adjustments to data models without requiring a predefined schema. MongoDB excels in scalability and performance, supporting horizontal scaling through sharding and replication for high availability. Its rich query language and indexing capabilities facilitate efficient data retrieval. Additionally, MongoDB integrates well with modern development frameworks and environments, making it a preferred choice for applications requiring rapid iteration, such as real-time analytics, content management, and IoT systems.

**Advantages:**

* Flexibility: Its schema-less design allows for easy modifications to the data model without disrupting existing data.
* Scalability: Supports horizontal scaling through sharding, enabling efficient handling of large volumes of data.
* Performance: Provides high-speed read and write operations with its in-memory processing and indexing features.
* High Availability: Includes built-in replication and failover capabilities to ensure continuous data availability.
* Ease of Use: Its document-oriented structure aligns well with modern programming practices and data structures.
* Rich Query Language: Offers a versatile query language that supports complex queries and indexing for efficient data retrieval.

**Disadvantages:**

* Lack of Transactions: Although recent versions have improved, multi-document transactions can be less robust compared to traditional relational databases.
* Consistency Issues: Being eventually consistent, it might not always provide immediate consistency, which can be problematic for some applications.
* Complex Queries: While it supports rich queries, complex aggregations can be less intuitive and performant compared to SQL databases.
* Memory Usage: High memory consumption due to its in-memory processing can be a concern for large datasets.
* Indexing Overhead: Maintaining multiple indexes can add overhead and impact write performance.

**2.3.** **Technologies used in this project.**

The HIGHTAIL platform is developed using the REPRA stack, which consists of React, Express.js, PHP, Relational Database (MySQL/Mongodb), and ASP.NET along with JavaScript, HTML & CSS. React Enables seamless and lightning-fast user interfaces, offering better UI rendering and performance compared to Angular. It allows users to design applications freely and organize code efficiently. Express.js Enhances productivity and streamlines web applications. PHP Works in conjunction with other technologies for backend processes. Relational Database (MySQL/Mongodb) Provides increased scalability and reliability. ASP.NET: Utilized for developing highly efficient, smoothly scalable, and easy-to-maintain websites. JavaScript Used for end-to-end development. HTML & CSS Essential for building the structure and styling of web pages.

**Chapter 3: Requirements and Analysis**

**3.1. Problem Definition**

Hightail, as an app for job seekers, addresses several critical challenges that individuals face in today’s competitive job market. One of the main issues is the difficulty of standing out to employers in a crowded field of applicants. Many job seekers struggle to present their skills and experiences in a way that catches the attention of recruiters or hiring managers. Hightail tackles this by offering a platform where users can create personalized, interactive portfolios, allowing them to showcase their work, projects, and accomplishments beyond a standard résumé.

Another challenge is the lack of direct communication between job seekers and potential employers. Often, job applications are submitted into systems where responses are delayed or non-existent. Hightail helps solve this by facilitating more direct interactions, enabling users to send their portfolios or applications directly to hiring managers with real-time feedback options. This makes the application process more dynamic and engaging for both parties.

Moreover, many job seekers struggle to maintain organization when applying to multiple positions, leading to missed opportunities or confusion. Hightail provides tools that help users manage their applications, track progress, and receive notifications on any updates, ensuring they stay on top of their job search. Additionally, the platform aids job seekers in networking by connecting them with professionals in their desired industry, making it easier to find mentors, collaborators, or even job referrals.

Overall, Hightail's goal is to simplify the job search process by offering a creative, organized, and interactive platform that helps job seekers present themselves effectively, stay connected with potential employers, and land their desired roles more efficiently.

**3.2. Requirements Specifications**

1. User Requirements

* Login or Register:
  + The user must be able to log in if they already have an account or register for a new account.
* Create Account:
  + If the user does not have an account, they should be able to create one by providing necessary details (e.g., username, email, password).
* Post User Content:
  + After logging in, the user should have the functionality to post content (user posts) that can be viewed by others.
* View Admin Posts:
  + Users should be able to view posts created by the admin after logging in and posting their own content.

2. Admin Requirements

* Login or Register:
  + The admin should be able to log in or create an account if not registered.
* Create Admin Account:
  + The admin can register by providing details like an admin ID and password.
* Post Admin Content:
  + After logging in, the admin should be able to post content (admin posts) that users can view.
* View User Posts:
  + The admin should be able to view posts made by users.

3. Conditional Logic Requirements

* Conditional Viewing:
  + The admin can only view user posts if the user has already posted content.
  + This condition ensures that admin access to user posts is dependent on the user activity (i.e., user posts first).

4. General System Requirements

* Authentication:
  + Both user and admin should have secure login credentials with encryption for password storage.
* Authorization:
  + Admins should have the capability to manage and view user-generated content, while users can only view admin content.
* Post Management:
  + Users and admins should be able to create, edit, or delete their respective posts.
* Access Control:
  + Admins should have higher access rights compared to users, with the ability to monitor and control user activity.

**3.3. Planning and Scheduling**

**3.3.1. Planning**

The planning phase for the Hightail application focuses on creating a comprehensive platform that connects job seekers with employment opportunities. The primary objective is to develop an application that allows users to post and view job-related content while enabling administrators to manage user activity and content effectively. The target audience includes job seekers looking for opportunities and employers wishing to post job openings. Key features for users include account registration, profile management, job-seeking content posting, and access to admin-posted jobs, while admins will have capabilities such as user management, job posting, and content moderation.

Hightail will function as an interactive platform designed to connect job seekers with employment opportunities, streamlining the job application process. Users will start by registering an account, providing essential information such as a username, email, and password, or logging in if they already have an account. Once logged in, users will access a personalized dashboard where they can post job-seeking content, view job listings from employers and the admin, and manage their applications. The interface for creating posts will be straightforward, allowing users to share details about their skills and experiences.

Admins will operate from a separate dashboard, equipped with enhanced functionalities such as user management, job post oversight, and analytics tracking to monitor engagement and posting frequency. A key feature will be the conditional access system, where admins can only view user posts if those users have made posts themselves, ensuring privacy and encouraging interaction. Hightail will also provide notifications to users regarding job postings, application updates, and messages, keeping them informed and engaged. Security measures, including password encryption and secure data storage, will safeguard user information, while support options like FAQs and contact forms will assist users with any issues they encounter. Overall, Hightail aims to enhance the job-seeking experience by fostering a collaborative environment for both job seekers and employers.

**3.3.2. Scheduling**

**3.3.2.1 Gantt Chart**

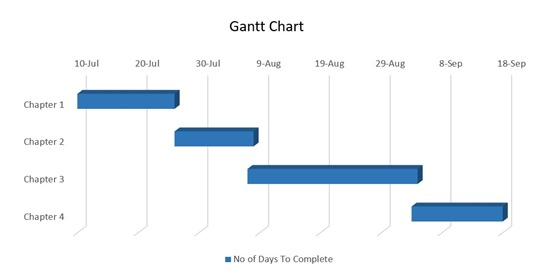
A Gantt chart is a horizontal bar chart used in project management to visually represent a project plan over time. Modern Gantt charts typically show you the timeline and status as well as who’s responsible for each task in the project. In other words, a Gantt chart is a super simple way to communicate what it will take to deliver a project on time and budget. That means it’s a whole lot easier to keep your project team and stakeholders on the same page from the get go. Reading a Gantt chart really comes down to understanding how the different elements come together to make a Gantt chart work.

To summarize, Gantt charts will allow you to:

* Know what the different activities are
* When activities need to start, and when they need to finish.
* How long your activities were primarily scheduled to last.
* How much your activities cost, and how much your project is likely to cost.
* Which activities depend on others to be effectively completed.
* Current progress status.
* The duration of the entire project.
* Know who is responsible for each task
* Find and take care of any problems associated with the project

The below Gantt chart shows the time taken to complete each chapter.



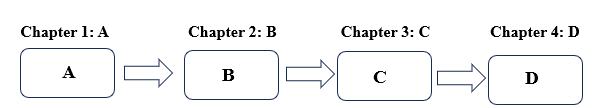


**Figure 3.1: Gantt Chart**

**3.3.2.2 PERT Chart**

A PERT chart is a visual project management tool that’s useful for mapping out project tasks and planning the overall project schedule. Many people confuse PERT with PERT chart, so the best way to provide a comprehensive PERT chart definition is to start with clarifying both of these terms. PERT stands for Program Evaluation Review Technique. PERT is the actual technique that is used to create a PERT chart. Meanwhile, a PERT chart is the visual diagram that results from using PERT. Think of PERT as the process and the PERT chart as the outcome. PERT charts allow project managers to see essential scheduling details such as task dependencies, task duration estimates, and the minimum amount of time a project can be completed within. But, they’re not the most user friendly or well understood tool.

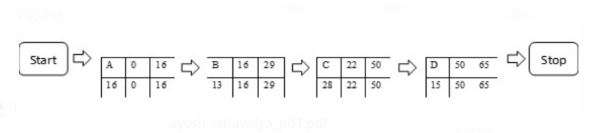
The below PERT diagram shows the schedule of the following chapters.

****

**Figure 3.2: Activity**

|  |  |  |
| --- | --- | --- |
| Activity | Precedence | Duration |
| A | - | 16 |
| B | A | 13 |
| C | B | 28 |
| D | C | 15 |

**Table 3.1: Critical Activity**



**Figure 3.3: Critical Activity**

Slack = 0 For all the Activities.

Therefore, Critical Path = A-B-C-D

**3.4. Software and Hardware Requirements**

**3.4.1 Software Requirements**

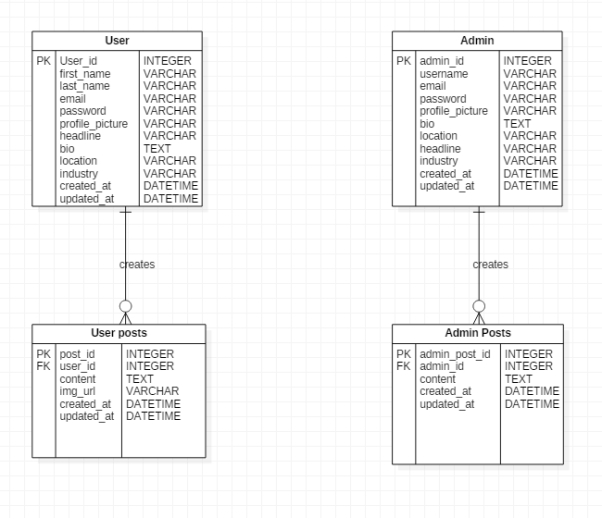
* Visual Studio
* MySQL / Oracle
* Canva
* Browser

**3.4.2 Hardware Requirements**

* CPU: Intel Core i5-8259U or higher
* RAM: 6GB or Higher
* Storage: 10GB
* Direct X: Version 11

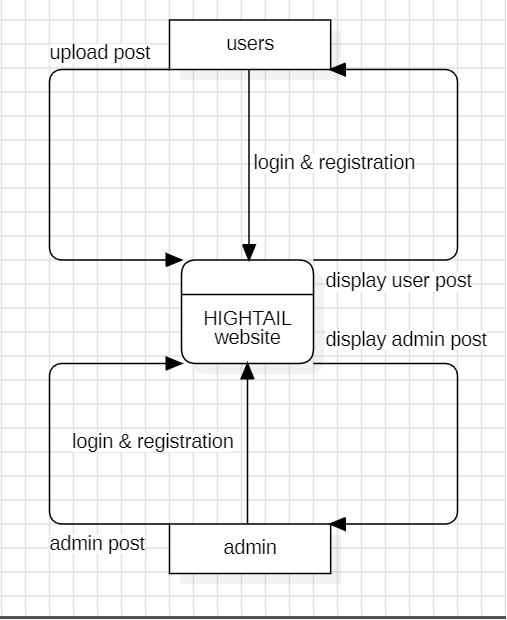
**3.5. Conceptual Models**

**3.5.1 Entity-Relationship diagram**

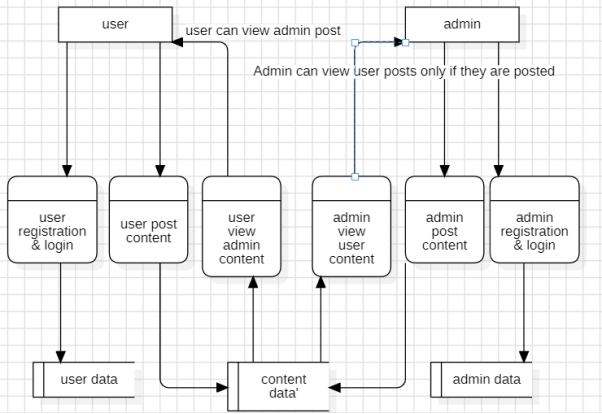


**Figure 3.4. Entity-Relationship diagram**

**3.5.2. Data Flow Diagram**

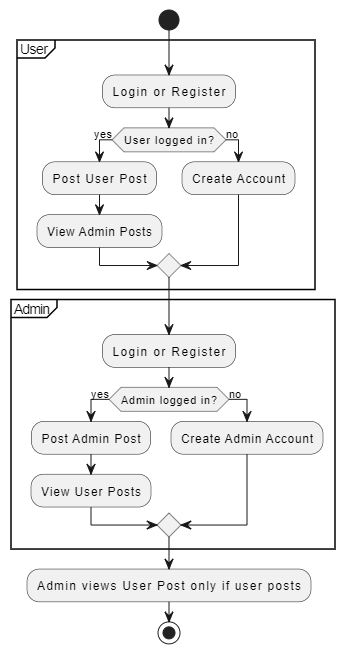


**Figure 3.5. Level 0 Data Flow Diagram**



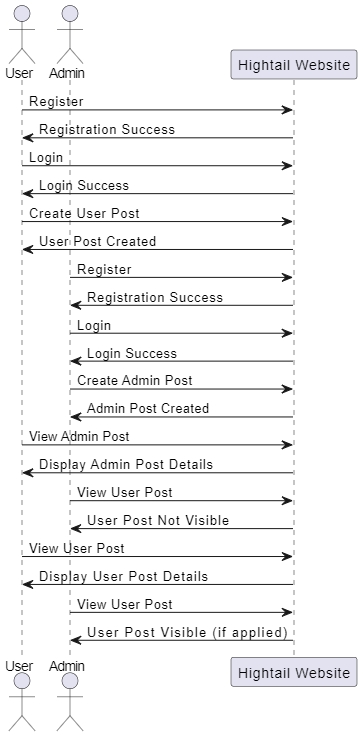
**Figure 3.6. Level 1 Data Flow Diagram**

**3.5.3 Activity Diagram**



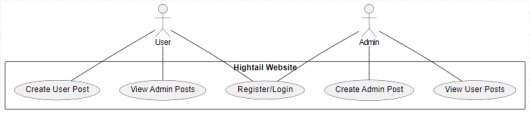
**Figure 3.7. Activity Diagram**

**3.5.4 Sequence Diagram**



**Figure 3.8. Sequence Diagram**

**3.5.5** **Use Case Diagram**



**Figure 3.9. Use Case Diagram**

**Chapter 4: System Design**

**4.1. Basic Modules**

The application will include the following modules –

•Homepage: homepage basic module includes a navigation bar with sections like profile , saved posts , search bar , categories displaying posts from connections ,profile views, and connection suggestions, enhancing user engagement and professional networking.

•Registration Module: This module is for those users who are new to this website. They must submit their details which will be stored in the database

•Login Module: This module will have a login form. The registered users will enter their login credentials like user id and password.

•User profile Module : It is a dedicated section on a platform that displays a user’s personal and professional information. It typically includes details like name, profile picture, bio and contact information, allowing others to view, connect, and interact with the user based on their shared interests or expertise.

•Admin profile Module: It is a section within a platform that displays an administrator's information and provides tools for managing the system. It typically includes details like the admin’s name, profile picture, contact information, and along with access to administrative functions such as user management, content moderation, and system settings. This module allows admins to oversee and control platform activities efficiently.

• User post Module: is a feature within a platform that allows users to create and share content, such as text, images, videos, or links, with their network or community. It typically includes options for composing posts, adding media , and interacting through likes, comments, or shares, enabling engagement and communication among users.

• Admin post Module: is a feature within a platform that enables administrators to create, manage, and publish content for the community or user base. This module typically includes tools for composing posts, adding multimedia, setting visibility options, moderating user comments, and scheduling posts, allowing admins to effectively communicate updates, announcements, or important information to users.

**4.2. Schema Design**

**4.2.1. User Schema**

|  |  |
| --- | --- |
| Column Name | Data Type |
| user\_id | INT |
| first\_name | VARCHAR |
| last\_name | VARCHAR |
| email | VARCHAR |
| password | VARCHAR |
| profile\_picture | VARCHAR |
| headline | VARCHAR |
| bio | TEXT |
| location | VARCHAR |
| industry | VARCHAR |
| created\_at | DATETIME |
| updated\_at | DATETIME |

**Table 4.1 User Module**

**4.2.2. Admin Schema**

|  |  |
| --- | --- |
| Column Name | Data Type |
| admin\_id | INT |
| username | VARCHAR |
| email | VARCHAR |
| password | VARCHAR |
| profile\_picture | VARCHAR |
| bio | TEXT |
| location | VARCHAR |
| headline | VARCHAR |
| industry | VARCHAR |
| created\_at | DATETIME |
| updated\_at | DATETIME |

**Table 4.2 Admin Module**

**4.2.3 User post Schema**

|  |  |
| --- | --- |
| Column Name | Data Type |
| user\_post\_id | INT |
| user\_id | INT |
| content | TEXT |
| created\_at | DATETIME |
| updated\_at | DATETIME |
| Image\_uri | VARCHAR |

**Table 4.3 User post Module**

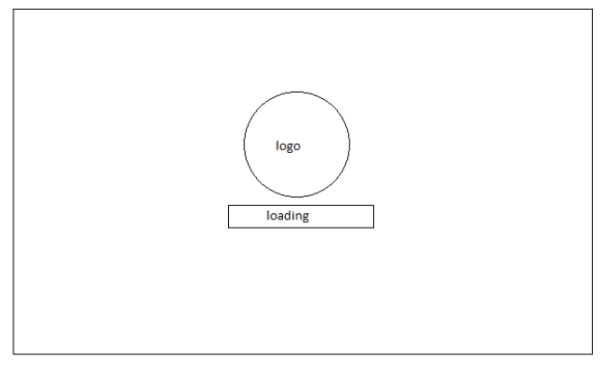
**4.2.4 Admin post Schema**

|  |  |
| --- | --- |
| Column Name | Data Type |
| admin\_post\_id | INT |
| admin\_id | INT |
| content | TEXT |
| created\_at | DATETIME |
| updated\_at | DATETIME |

**Table 4.4 Admin post Module**

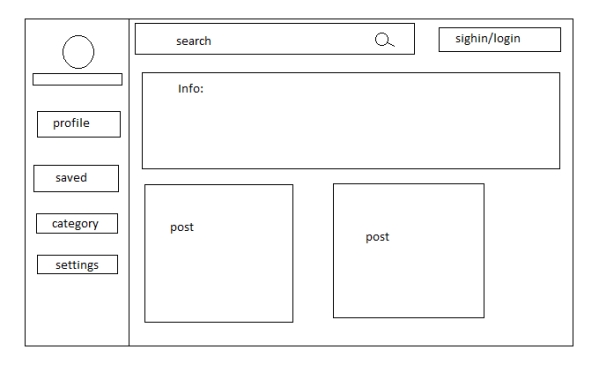
**4.3 User Interface Module**

**4.3.1 Loading Page**



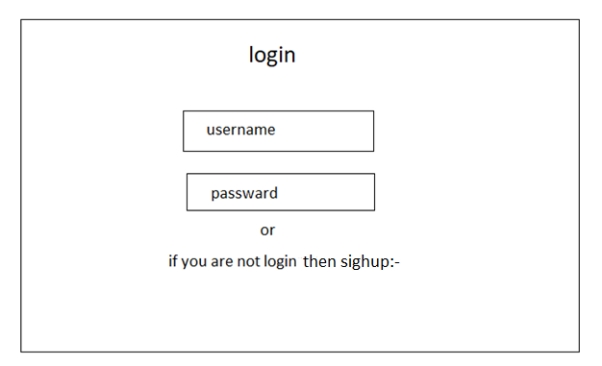
**Figure 4.1. Loading Page**

**4.3.2 Home Page**



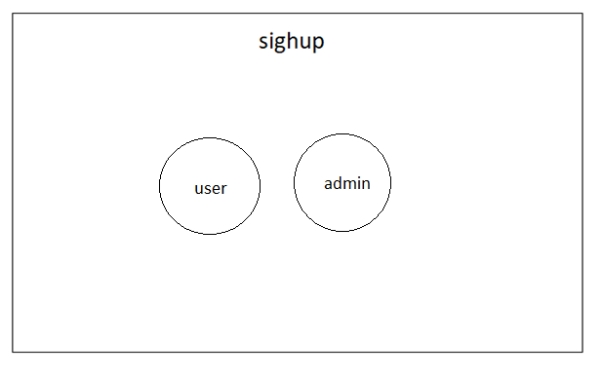
**Figure 4.2. Home Page**

**4.3.3 Login Page**



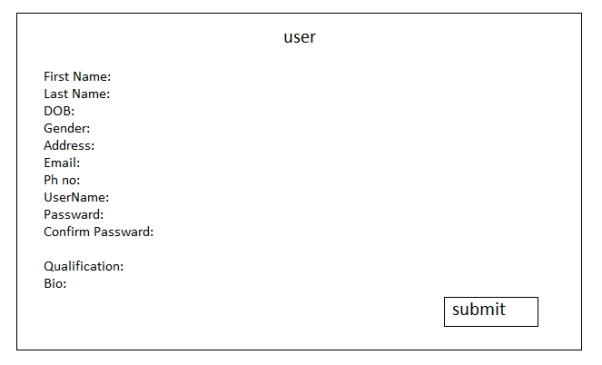
**Figure 4.3. Login Page**

**4.3.4 Sign up Page**



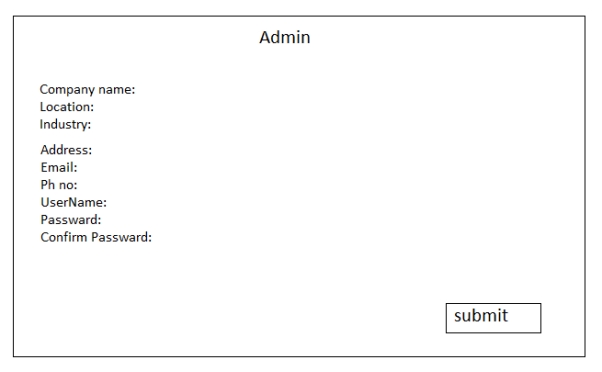
**Figure 4.4. Sign up Page**

**4.3.5 User Sign up Page**



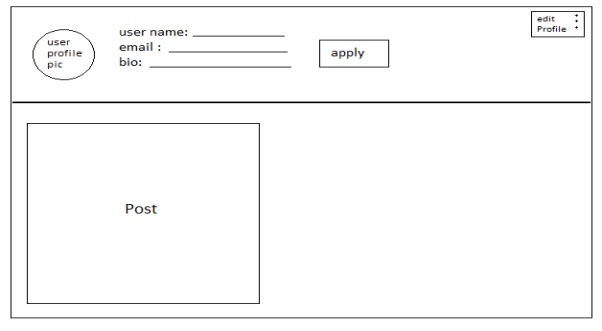
**Figure 4.5. User Sign up Page**

**4.3.6 Admin Sign up Page**



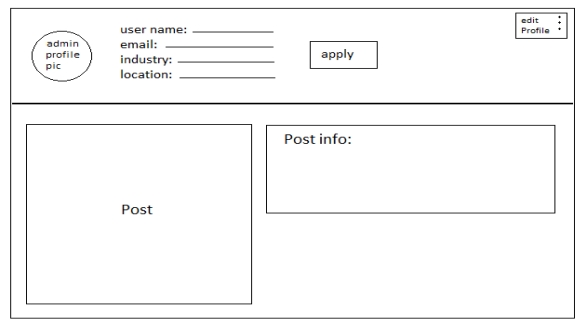
**Figure 4.6. Admin Sign up Page**

**4.3.7 User Profile Page**



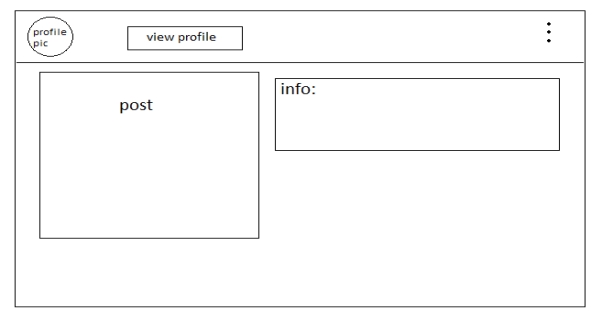
**Figure 4.7. User Profile Page**

**4.3.8 Admin Profile Page**



**Figure 4.8. Admin Profile Page**

**4.3.9 Post Page**



**Figure 4.9. Post Page**

* 1. **Security Issues**

HIGHTAIL, a job search platform for aspiring professionals, must prioritize security in handling user data, including email addresses and contact details. While the site does not facilitate payments, the collection of personal information poses risks if not properly secured. Users should be vigilant about the data they share and ensure that HIGHTAIL employs robust encryption and security protocols to protect sensitive information. Back-end measures, such as secure data storage and regular security audits, are essential to prevent data leaks and unauthorized access. By maintaining transparent data handling policies and implementing strong security measures, HIGHTAIL can create a safe and secure job search experience for all users

**References**

[**https://www.geeksforgeeks.org/html-introduction/**](https://www.geeksforgeeks.org/html-introduction/)

[**https://www.w3schools.com/sql/**](https://www.w3schools.com/sql/)

[**https://www.wonsulting.com/blog/top-10-problems-job-seekers-have-and-how-to-solve-them**](https://www.wonsulting.com/blog/top-10-problems-job-seekers-have-and-how-to-solve-them)