**Analysis of Needs**

Analyzing the requirements for the requirements bus system involves identifying and analyzing the features and functionality needed for the web application to meet the intended goals. Here are the key requirements for a typical bus claim system:

* **User Interface:** The system should have a user-friendly interface, allowing users to easily search for available bus routes and connections between specific start and end points. Also,
  + Register new routes.
  + Delete existing routes.
  + Query routes and interchanges by giving start and end points.
* **Route database:** The system must have a complete database of all available bus routes and their respective starting and ending points, as well as connections and any relevant information about routes, fares and timetable.
* **Real-time updates:** The system is expected to provide real-time updates on bus journey times, delays and cancellations to ensure users can make informed decisions about their journeys.
* **Security**: The system must be secure to prevent unauthorized access and protect user information such as payment details and login information.
* **Payment integration:** The system should incorporate a payment gateway to allow users to pay for tickets online.
* **Reporting and Analysis:** The system will generate reports and analytics to give bus companies insight into the most frequently used routes, passenger numbers and revenue reports.

Byidentifying and analyzing these requirements, I can ensure that the bus request system is designed to meet the needs of the user and efficiently provide the requested functionality.

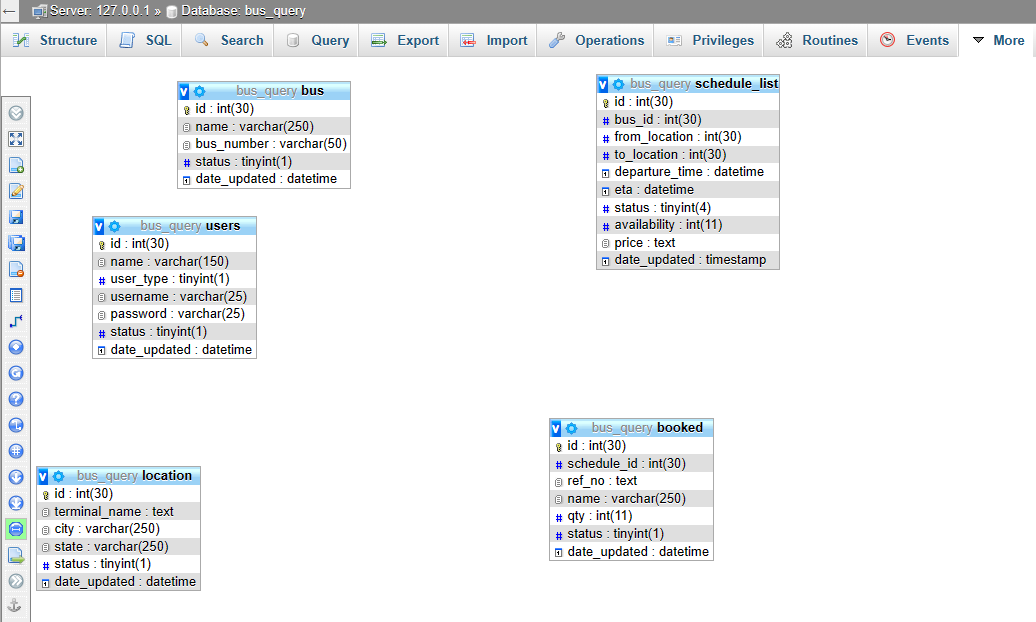
|  |
| --- |
| The UI design of the Bus Inquiry System should emphasize an intuitive and user-friendly experience, ensuring ease of use and accessibility, as well as an aesthetic and operationally responsive layout. good on different types of devices and browsers. The main areas of focus for user interface design include the main pages of the system, designed to engage users and maximize their user experience.   * **Home Page:** The Bus Query System's home page acts as the gateway for users and must exhibit an overview of the application's key features. It must offer a responsive and easy-to-use form where users can enter their desired destination and travel date in a user-friendly manner. * **Admin Portal:** The admin panel of the Bus Query System is a confidential segment reserved for authorized administrators, which features a secure login interface to ensure access only to authenticated administrators. Once in, administrators can use the user-friendly and intuitive interface designed for efficient route management tasks, including the addition, updating, and deletion of routes, among others.     **System Design** |

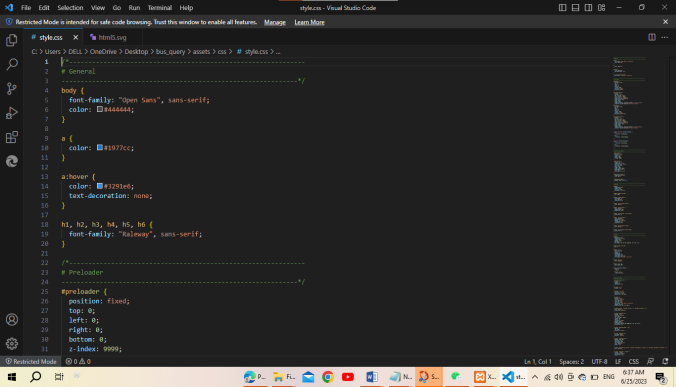
**Design Back-End Features**

The architecture of the backend The query bus system plays an important role in clearly handling user requests, managing the database, and ensuring a smooth flow of data to the user interface. use. This backend infrastructure consists of several critical components that work together closely to ensure optimal system functionality. These components include:

* Handling user input: Using PHP scripts, user searches are validated and handled appropriately. Processed data is passed to the appropriate functions to perform the required processing.
* Database management: MySQL, a reliable relational database management system, stores and manages important information about bus routes and dates. Database schema design focuses on achieving efficient and efficient data storage, retrieval, and processing. SQL queries perform adding new routes, updating existing routes, deleting routes and retrieving the route information requested by the user.
* Make a routing request: Initiating a user search will submit the system to perform a database search necessary to retrieve associated bus routes and intersections and provide fast-responsive search results. This polling mechanism must be optimized to achieve fast response times while dealing with large data sets.
* Authentication and authorization: The system has a secure login mechanism to authenticate the administrator before accessing the admin panel. Appropriate authentication, authorization, and access control measures are in place to restrict access to functions and data based on administrator roles and permissions.
* Authentication and data integrity: Full data validation ensures user input matches defined standards and consistency. Validation checks include many differences in the input field, such as date format and destination name. Furthermore, the system enforces data integrity constraints to maintain the regularity and reliability of stored data.
* Error handling and logging: A well-configured error handling mechanism ensures that the bus request system handles rare exceptions and errors more gently. Errors and critical information will be displayed in a user-friendly approach, allowing for easy troubleshooting. Logging relevant events and errors is essential for future traceability, debugging, and analysis.

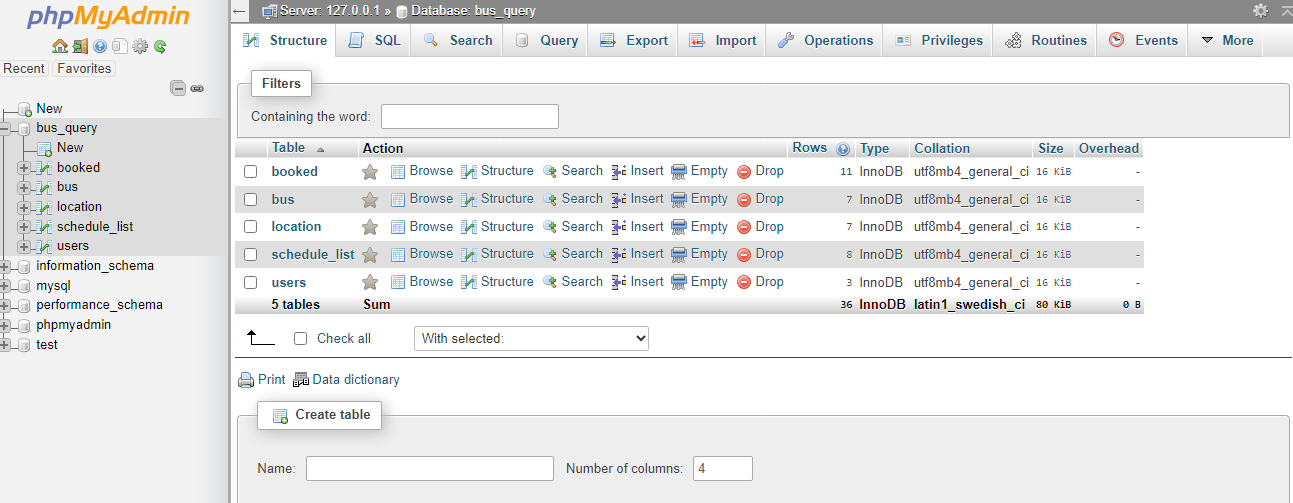
|  |
| --- |
| **Implementation** As the Bus Query System moves into the implementation phase, its design undergoes practical transformation into a fully functional system. This phase involves developing code, designing a workable database, integrating different components, and conducting comprehensive testing of the overall system. The implementation process can be broken down into the following critical aspects.  **STEPS 1**   * To set off the implementation phase of the Bus Query System, I set up and configured a local development environment using XAMPP v7.2.1-0-VC. This package bundles Apache HTTP Server, PHP, and MySQL, creating an ideal platform for coding and testing the application. I further configured PHP and MySQL settings, including database connection parameters and file upload limit. These tweaks were critical in ensuring that the development server runs smoothly, guaranteeing seamless testing and debugging of the Bus Query System.     Implementing a bus query system utilizes a set of technologies specifically chosen to provide an efficient and seamless user experience. These include:   * **PHP:** Server-side scripting languages ​​are widely used for processing user requests, interacting with databases, and manipulating data for various functions within bus query systems. * **MySQL:** The relational database management system of choice as the primary storage technology for all bus route and date information. * **HTML/CSS**: An important tool used to create interactive interfaces. HTML is responsible for structuring and defining layout, CSS is responsible for styling. * **JS File:** A must-have component for optimizing user engagement, enhancing interactivity, and providing a great user experience.   Together, these technologies provide a robust, efficient, and reliable capability that is essential for a successful query system and I succefully create my project with this thing. |





**STEPS 3**

* For UI development, I created HTML/CSS templates for various pages of the system. Home page, search results page, admin panel. Ensure a seamless visual experience. Additionally, we used JavaScript to extend the user interface to provide interactive elements, form validation, and dynamic content loading. All this is aimed at improving user interaction.
* Backend development included implementing PHP scripts to process user requests and process form data while connecting to the database. To enhance the functionality of the system, we have created several key functions for managing user input validation, database queries, route management and user authentication functions.



**STEP 2**

* I began by designing an optimal database schema to store relevant data, such as bus routes, schedules, interchanges, and administrator details. This involved creating necessary tables in the MySQL database with appropriate data types, ensuring proper relationships and constraints. To jumpstart the dataset for the system, I implemented scripts to add initial data, such as predefined routes and sample administrator accounts.

|  |  |  |
| --- | --- | --- |
| Key Findings | | |
| Key Findings #1 Research and argument  [To replace a photo with your own, just delete it and then, on the Insert tab, click Picture.] | Key Findings #2 Research and argument | Key Findings #3 Research and argument |

|  |
| --- |
| **STEPS 4**  During the development phase of the bus query system, I focused on important aspects such as management panel development, application testing and deployment.   * Leveraging our expertise in authentication and session management, i have created a management panel to help users effectively manage theirr bus routes. Additionally, i have integrated the appropriate web forms and controls needed to add, update, and delete bus routes. Management features make it easy to add routes, update route details, and delete routes, further improving the efficiency of the system. * During the integration and testing phase, i created a seamless communication channel between our front-end and back-end systems. i tested different user scenarios covering important areas such as finding routes, adding new routes via the admin panel, and validating user authentication and authorization. I also fixed bugs and discrepancies that occurred during testing. * Finally, I deployed the Bus Query system to production servers and configured the environment to ensure broad access. This allowed us to increase security by setting up the proper server configuration, enabling system logging and monitoring, and ensuring effective performance optimization. I constantly monitor and maintain the system and provide the necessary updates, security patches and performance enhancements to keep the system running smoothly. |

**TESTING**

As part of the development process of the Bus Query System, I carried out several testing procedures to ensure its reliability and functionality.

* Unit testing was employed, where each component of the application was tested individually and in isolation to affirm its correctness and functionality.
* Next, Integration testing was done, where the interaction between different system components was tested to verify proper communication, data flow, and integration.
* The entire system was also subjected to System tests that evaluated its behaviour and functionality, validating both user-facing features and administrative functionalities.
* Performance testing was also carried out, using load conditions to evaluate the application's performance under different load conditions and ensure it can handle a significant number of concurrent users without performance degradation.
* Security was also a vital consideration, and as such, the Bus Query System was tested for vulnerabilities, weaknesses, and protection against common security threats.
* User Acceptance testing was also conducted, with the participation of real users or test participants, to assess the system's usability, user-friendliness and overall user satisfaction.
* Furthermore, the test cases were designed based on identified requirements, covering various functionalities, edge cases, and potential error conditions. Results were recorded, and any identified issues were addressed and resolved before deployment.

Finally, a testing demonstration video was included with the project, demonstrating the functionality and capabilities of the Bus Query System, allowing for better understandability of the system.

# **Conclusion**

The Bus Query System is an advanced web-based platform designed to give users efficient and hassle-free access to bus route information. I attaches great importance to user experience, ensuring that the systems I develop are user-friendly, easy to operate, and highly functional.

I have performed a thorough requirements analysis to understand the needs of our users, and have integrated functions such as creating new routes, deleting existing routes, and querying bus routes. During the system design phase, I defined the system architecture, designed an intuitive user interface, created robust backend functionality, implemented top-notch security measures, and managed the database. The platform is implemented using PHP, MySQL, HTML/CSS, and JavaScript, resulting in a stable, responsive, and feature-rich system.

Rigorously tested to prioritize system reliability, performance and security. My main goal is to provide a seamless solution that allows users to search for bus routes and allows authorized personnel to manage bus routes effectively. My goal is to improve the overall user experience by making accessing bus route information easy, seamless and efficient.