**LAB AUTOMATION SYSTEM**

**Participants:** Muhammad Yousuf, Khuzaima Ahmed, Shoaib Shah, Muhammad Yahya

#### Table of Contents

Introduction …………………………………………………………………………02

Project Synopsis……………………………………………………………………..02

Project Analysis …………………………………………………………………….02

Project Design ….…………………………………………………………………..03

DFDs…….…………………………………………………………………………..04

Process Diagram……………………………………………………………………..04

Screen Shots ………………………………………………………………………..05

Source Code ………………………………………………………………………..05

Users Guide ………………………………………………………………………..05

Developer’s Guide …………………………………………………………………..06

***Introduction:***–

The Lab Automation System is a software application designed in collaboration with the aforementioned team members to automate laboratory or testing facility operations. The primary objective of this system is to streamline testing processes, facilitate data management, and enhance overall productivity. It features a user-friendly interface that enables easy navigation for users.The aim of this project is to provide a comprehensive lab automation solution that monitors testing processes, maintains records, generates reports, and enhances overall laboratory management.

***Project Sysnopsis:***–

The primary aim of this project is to provide automation for the lab system. It was noticed that maintaining records in large registers was becoming increasingly difficult, with occasional errors in data entry or misplacement of records. Therefore, there was a need for automation to prevent manual record-keeping in registers and to store records efficiently in databases. The project was developed to address this need, enabling automation for the lab system and facilitating the storage of records in databases. With the Lab Automation System, users can fetch records using advanced search functionality by entering testing ID, product ID, or product code. They can also view the details associated with each record with just one click. Additionally, users can search for records solely based on testing ID or product code, and even by product name to view all products associated with that name. Furthermore, the system enables efficient testing of products and provides features for generating reports, conducting analysis, and managing users. To use the system, users need to create an account on the Lab Automation System and then log in. If users are not administrators, they will have access to the Main Page, Product Testing Page, Records List, and Product Information Page. They can test products, search for records using advanced search, and view the current list of records. They can also fetch detailed information about a product using only the product code or testing ID. If users are administrators, they will have access to all seven pages of the site, including the Main Page, Product Testing Page, Records List Page, and Product Information Pages, as well as the Reports Page, Financial Management Page, and User Management Pages. Administrators can generate reports summarizing the testing process and download them. They can also conduct analysis and view the number of registered users, administrators, and regular users. Additionally, administrators can use advanced search to find specific users by name, email, or contact number. They can edit or delete user accounts and product details. Overall, the Lab Automation System offers several unique and effective features that make it user-friendly and easy to use.

***Project Analysis:***–

The Lab Automation System is an experimental project designed to automate the operations of a laboratory. The purpose of this system is to assist in making laboratory processes efficient and reliable. Through this system, the aim is to make daily tasks easier for users of any laboratory.

**Features:**

**Record Management**: There is no need to maintain records in large registers in the Lab Automation System because records are stored in databases. This system helps in organizing and accessing records.

**Search Functionality**: With advanced search, any record can be easily found by entering testing ID, product ID, or product code. This provides users with specific information quickly and accurately.

**User Management**: In this system, users are given different roles and permissions. Admins have full access to the site, while regular users have limited access.

**Testing and Reporting**: Products can be tested more effectively in the Lab Automation System. Reports can be generated summarizing the testing process, and these reports can also be downloaded.

**Analysis**: Data analysis can also be performed through the system, allowing users to understand the performance and trends of the lab.

**Security**: The data of every user is kept secure. Access control and authentication mechanisms ensure that only authorized users can use the system.

***Project Design:***–

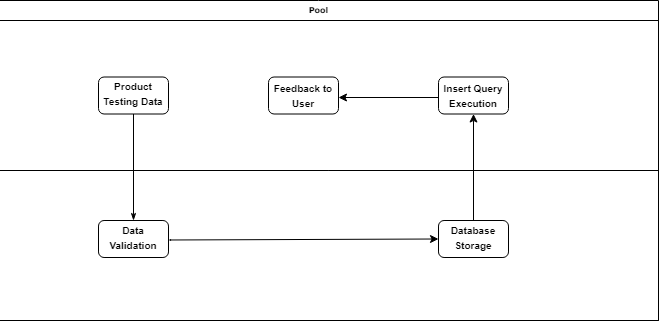
In the Project Design phase, we put a lot of effort into crafting the Lab Automation System. Our primary goal was to make the system as user-friendly as possible, and our efforts proved successful. We initially devised several structures, but during implementation, we had to make numerous changes. We continuously modified the system's architecture, removing elements and incorporating new features.

Even after finalizing the structure, significant changes were made. Our aim was to ensure that whenever a user opens the web application, they encounter the authentication page first. Additionally, we kept the UI simple, with the Navbar positioned at the top. The Navbar includes the Lab Automation logo on the left side and page links in the center. On the right side of the Navbar, there's a Username accompanied by a User Icon, which also serves as a dropdown menu. Clicking on it provides two options: Settings and Logout.

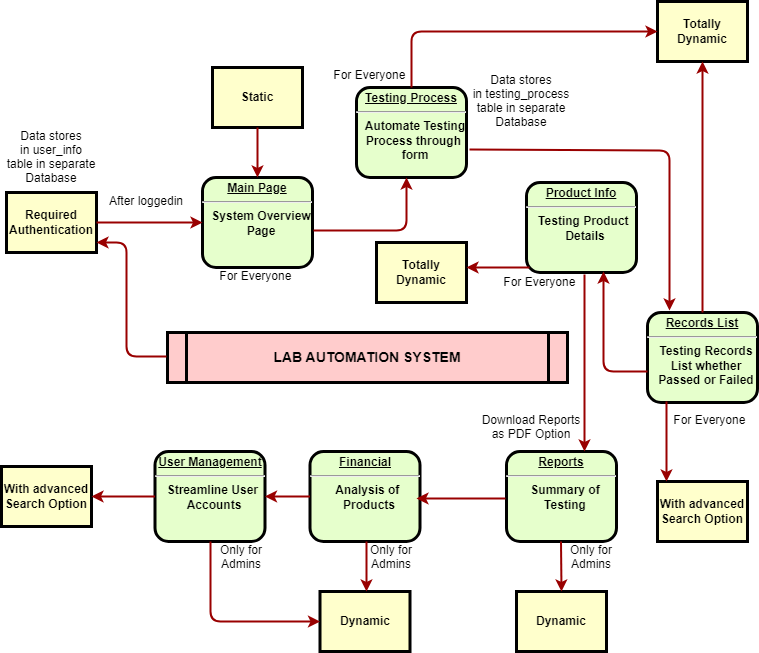
A Footer is provided below each page, and the content of each page is situated between the Navbar and Footer. The main page is static, while all other pages are dynamic. For data storage, we utilized MySQL databases and employed simple queries for data storage on the backend. Generating reports is also straightforward, utilizing fetch assoc array. For downloading reports in PDF format, we utilized fpdf 1.8.3 on the backend.

We minimized custom styling throughout the project, primarily relying on Bootstrap. jQuery was used for AJAX requests, and data tables were employed for advanced search functionality. Our overarching aim was to make the Lab Automation System as user-friendly and easy to use as possible.

***DFDs:***–



***Process Diagram:***–



***Screen Shots:***–

A folder named **'Screen Shots'** has been created in the Documentation Directory, containing high-quality screenshots of every page within the Lab Automation System.

***Source Code:***–

The source code, accompanied by clear and concise comments, is available in the PHP Lab Automation folder for easy reference.

***User Guide:***–

When the user opens the site, the first page they will encounter is the Authentication page (Login Page). They will need to create their account first and then log in to access the main page. The main page, as described in the Project Synopsis, is a static page where the user will find a Brief Overview of the site along with Quick Links such as Product Information page, Product Testing page, Product Records pages, and a Contact Section. The Navbar contains links to all pages.

If the user wants to test a product, they can click on the Product Testing page, fill in the product details, and click on the Add Product button. The data will be stored in the testing\_process table if the user has filled in the details correctly and all required fields are filled. If any required field is left empty, an error will occur. Similarly, if the Product ID exceeds 10 digits, an error will occur. The Testing ID will be an auto-generated 12-digit code.

After adding a product, the user can go to the Records List page to view all records. They can also perform an advanced search and edit or delete records if they are an Admin. However, only necessary information will be displayed on the record list.

If the user wants to edit their account, they can click on their username in the Navbar, where a dropdown menu will appear with options for Logout and Settings. From there, the user can update their account details and change their account password.

Beyond this point, only Admins have access. If an Admin wants to view Testing Records reports, they can go to the Reports page. By providing Start Date and End Date, the Admin can generate reports. If records are available for the provided dates, the data will be fetched and displayed in a table on the right side. The Admin can also download reports in .pdf format.

If an Admin wants to view Analysis, they can go to the Financial Management page and click on View Analysis to see the Testing Summary. If an Admin wants to remake a product, they can submit the Testing ID and Product ID for remaking and mark it accordingly.

If an Admin wants to see the total Registered Users on the site, they can go to the User Management page. There, they will find data for all registered users. They can filter users by roles, edit user account details, delete user accounts, and perform an advanced search.

This is a comprehensive User Guide designed to be easily understood and followed.

***Developer’s Guide:***–

**Introduction to Development Environment:**

The Lab Automation System is developed using PHP for the back-end logic, HTML, CSS, JavaScript, and Bootstrap for the front-end interface, and MySQL for the database management system.

To set up the development environment, ensure that you have PHP, MySQL, and a web server (such as Apache or Nginx) installed on your system.

Clone the project repository from the [Project link](https://github.com/YousufNaveedKhan/PHP-Lab-Automation-System.git) provided.

Import the database schema using the SQL dump file provided in the project repository.

Configure the database connection settings in the appropriate configuration file.

**Architecture Overview:**

The Model represents the data and business logic, the View represents the user interface, and the Controller handles the user input and interactions.

**Folder Structure:**

The project folder structure is organized as follows:

assets: Contains CSS, JavaScript, and image files.

assets/php: Contains PHP files for backend logic.

assets/php/connections: Contains PHP files for database connections.

assets/php/auth.php: Contains authentication logic.

assets/php/navbar.php: Contains code for the navigation bar.

assets/php/footer.php: Contains code for the footer section.

documentation: Contains documentation files, including user guides and developer guides.

**Coding Standards:**

Follow [PSR (PHP Standards Recommendations)](https://www.php-fig.org/psr/) for PHP coding standards.

Use consistent naming conventions for variables, functions, and classes.

Includes comments in the code to explain the purpose of functions, classes, and complex logic.

**Source Code Overview:**

Main source code files include main.php, login.php, signup.php, product\_testing.php, testing\_records.php, product\_information.php.

Each file handles specific functionalities such as authentication, user registration, product testing, and record management.

Comments within the source code provide insights into the functionality of each section.

**Database Schema:**

The database schema consists of tables such as user\_info, testing\_process, etc.

Import the database schema using the provided SQL dump file (lab\_automation.sql, testing\_process.sql).

Configure the database connection settings in assets/php/connections/connection.php and assets/php/connections/testing\_connection.php.

**API Documentation:**

Currently, the Lab Automation System does not include external APIs.

**Testing Procedures:**

Testing procedures include manual testing of each functionality to ensure proper functionality and user experience.

Developers can use tools like PHPUnit for automated unit testing.

**Deployment Instructions:**

Deploy the Lab Automation System to a web server with PHP and MySQL support.

Ensure proper file permissions and security measures are implemented.

Update configuration files with production environment settings.

**Contributing Guidelines:**

Fork the project repository, make changes in a separate branch, and submit a pull request for review.

Follow coding standards and guidelines provided in the Developer's Guide.

Participate in code reviews and discussions for improvements.

**Troubleshooting and Support:**

For technical support and issue resolution, refer to the project's GitHub repository or contact the project maintainers.

Refer to the documentation for common issues and solutions.

**Version Control:**

The project uses Git for version control.

Clone the project repository, create feature branches for development, and merge changes using pull requests.