Technical Documents on the Web Application

In this web application, I've identified several significant components, each with specific roles, and they collaborate to create a functional system.

* 1. Database:

One of the main components used in this project is the implementation of php, therefore for the database I used the phpmyadmin to create and insert data and store it there. The raw data is the same as the previous project

* 2. Database Connection (database.php): To interact with the database, I need to address a connection between the web page and the database. By writing the localhost, name, password, username, those are the components to simply connect with the database.
* 3. Sessions: By creating ‘session\_start’, this helps to start running the php connection.
* 4. User Authentication (login.php): The login page will serve as the base of the whole webpage, to start things off. In the login page, the imputed username and password will be fetched (by using the select function of mysql) and later on checking it to see whether it matches what i have in the database, once they are a match, users are immediately directed to the welcome page.
* 5. Main Application Page (varies): The primary application page varies depending on the provided code. Typically, it's where users land after successful login. This page displays dynamic content, such as student data, course information, grades, etc. It employs SQL queries to retrieve data from the database and PHP for processing and presentation. It frequently includes tables to present data to users.
* 6. URL Parameters: Some sections of the application use URL parameters to specify specific details. For instance, the 'population.php' and 'grade.php' pages utilise URL parameters to indicate the program, year, and course. PHP code processes these parameters to retrieve the pertinent data from the database.
* 7. Form Handling (Add Student, Add Course, Edit Grade): Across the application, various forms allow users to add students, courses, and edit grades. When a user submits a form, the data is dispatched to the server, where PHP scripts take charge of processing it. These PHP scripts validate, sanitise, and insert/update data within the database. Following successful operations, users are often redirected to relevant pages.
* 8. Error Handling: The application incorporates mechanisms for error handling. It can exhibit error messages, such as when a user enters incorrect login credentials or when a database query encounters an issue.
* 9. Navigation: Navigation is facilitated through hyperlinks and buttons, allowing users to move between different sections of the application, such as viewing student populations, adding students, checking grades, and more.

In summary, this web application comprises a database, a database connection, user authentication, multiple web pages serving various functions, form handling, HTML/CSS for the user interface, JavaScript for enhancing user interactions, error handling, and robust security measures. These components collaborate seamlessly to provide users with an intuitive platform for managing student data, courses, and grades. The PHP scripts act as the logic that binds everything together, enabling data retrieval, processing, and interaction with the database.