**CODE SIMPLE EXPLANATION**

**Class:** classes are the building blocks of object-oriented programming, enabling code reusability, modularity, and a more intuitive representation of real-world entities and interactions in software systems. Here are the classes that is used in making the system:

* user\_log\_up.php - This PHP code is responsible for displaying a table with log data retrieved from a MySQL database. The table contains various columns such as ID, Name, Serial Number, Card UID, Device Department, Date, Time In, and Time Out. It seems to be a part of a web application where users can filter and view logs based on certain criteria.
* UsersLog.php - This class generates an HTML page to display user logs. It ensures that only logged-in administrators can access the logs. The page has a navigation bar and a main section showing the logs. Users can filter the logs by date, user, and department using a filter button. They can also export the filtered logs to Excel.
* ac\_login.php - this class is responsible for validating the admin's login credentials against the database and setting up a session upon successful login, allowing the user to access restricted parts of the website. If there are any errors during the login process, the user is redirected back to the login page with corresponding error messages.
* ac\_reset\_pass.php - this class is responsible for handling password reset requests, verifying the validity of the requests, updating the user's password, and providing appropriate error messages or success messages during the process.
* ac\_update.php - This class helps maintain and update the admin's information in the system.
* connectDB.php - This PHP class connects to a MySQL database named "rfidattendance" running on the localhost server using the username "root" and an empty password. If the connection fails, it shows an error message. Otherwise, it creates a successful database connection that can be used to interact with the database in the rest of the PHP script.
* devices.php - this class sets up a web page for managing devices, checks if the user is logged in as an admin, and allows the admin to add new devices. It also loads data related to devices and user logs dynamically using AJAX.
* dev\_config.php - This class interacts with the database to manage device-related information and user account details in the web application.
* dev\_up.php - This PHP class creates a table showing device information fetched from a database. Each row in the table represents a device and displays its name, department, UID, date, mode, and a delete button. The 'mode' can be selected using radio buttons, and the 'UID' can be updated using a refresh button. The 'delete' button removes the device from the database.
* Export\_Excel.php - This class generates an Excel file containing user log data based on certain filters. It checks if a form with a "To\_Excel" button is submitted. If yes, it retrieves filter values like start date, end date, card selection, and device selection. Then, it fetches data from the database based on these filters and creates an Excel file with the results. If there are no matching records, it redirects the user to a log page.
* footer.php - this code represents a basic layout for a website footer with three sections: "Contrast," "About Us," and "Contact Info." The footer contains relevant information about the website and its contact details, while the copyright section provides the copyright notice and some additional information. It's important to note that this code snippet is just a part of the whole website, and the entire website might have additional HTML, CSS, and JavaScript to make it functional and visually appealing.
* getdata.php – this code/class handles user login, logout, and RFID card registration operations, and it communicates with a MySQL database to store and retrieve the relevant data.
* header.php – the code is a mix of frontend (HTML, CSS, and JavaScript) and backend (PHP) elements, enabling a basic contact page with account update functionality for administrators.
* index.php - this web page is a user dashboard that displays a list of users with specific information from a database. The users must be logged in as an admin to access this page, and it provides navigation links to other related pages. The page also includes some interactive elements, such as the navigation menu and a loader while the content is loading.
* install.php – This class/code handles the creation of the database and tables and displays appropriate messages indicating whether these operations were successful or if there was an error during the process. Keep in mind that this code is primarily focused on creating the database and tables, but it does not include any data insertion or retrieval functionality. You would need additional code to interact with these tables, such as inserting user records, logging attendance, or fetching data from these tables.
* login.php - This code is for a web application related to a library attendance tracking system using RFID technology. It checks if the user is logged in and redirects them to the login page if not. The page has a banner and sections that describe the system's features. It includes a login modal with error messages, a navigation menu, and some JavaScript for various functionalities.
* Logout.php - this code sequence effectively clears the session data for the current user and logs them out by destroying the session. After that, it redirects the user to the "login.php" page.
* ManageUsers.php - This code represents a web page that manages user information. It checks if the user is logged in as an admin; if not, it redirects them to the login page. The rest of the code creates an interface to manage users, including fields for user details, a department dropdown, and buttons to add, update, and remove users.
* manage\_users\_conf.php - The code handles user-related operations like adding, updating, and deleting users in a database.
* manage\_users\_up.php - this code fetches data from a MySQL database, creates an HTML table, and displays the data along with buttons for each UID. It also adds some conditional formatting to indicate the selected UID. The actual functionality of selecting a UID and performing subsequent actions is not shown in this code snippet and would require additional JavaScript or backend processing.
* new\_pass.php - this code is part of a password reset functionality that uses a combination of URL parameters (selector and validator) to allow users to reset their passwords securely. If the URL contains valid selector and validator parameters, the code displays a password reset form to the user, and upon submission of the form, the ac\_reset\_pass.php script should be responsible for handling the password update process.
* reset\_pass.php - this code allows users to request a password reset by providing their email address. It generates a unique URL with a token and selector, stores the request in the database, and sends an email to the user containing the password reset link. The user can click on the link to access the password reset page where they can set a new password.
* rfidattendance.sql - this code serves as a script for creating the database schema and initializing it with some initial data for an RFID attendance management system. The system seems to involve administration, devices management, user information, and attendance logs for tracking user check-ins and check-outs using RFID cards.

**CSS (General Idea on what is the function of those indicated CSS in your system):** is a crucial technology used alongside HTML to style and format web pages. It allows web developers to control the presentation and layout of HTML elements, including properties like colors, fonts, sizes, margins, and padding. By separating the content layer from the presentation layer, CSS makes it easier to maintain and update websites without altering the underlying structure. It enables responsive design for different screen sizes and devices, supports animations and transitions, and enhances accessibility. CSS also allows customization and theming, empowering developers to create visually appealing and user-friendly interfaces for websites and web applications.

**JSQuery (.js files)**: jQuery is a fast, lightweight, and popular JavaScript library designed to simplify client-side scripting in web development. It provides a wide range of functions and utilities that make it easier to manipulate HTML documents, handle events, perform animations, and make AJAX interactions. jQuery abstracts away many of the complexities and inconsistencies of native JavaScript across different web browsers, allowing developers to write more concise and cross-browser compatible code.