

Login System Project Documentation

Overview

This document provides a detailed description of the files and functionalities in the Login System project. Each file is annotated with its purpose, functions, parameters, and return types.

File Structure

```
/login-system
├── index.html           # Frontend HTML page for user login and
registration
├── style.css           # CSS styles for a modern UI
├── script.js           # JavaScript for AJAX calls
├── api/
│   ├── index.php       # Main API endpoint with routing
│   ├── UserService.php # Service class to handle user logic
│   ├── db.php          # Database connection file
│   └── routes.php       # Routes management
└── README.md           # Instructions on building and running the project
```

File Descriptions

1. index.html

Purpose: The main frontend file for user interaction.

Key Functionalities:

- Provides a login form.
- Allows users to register.
- Includes a "Remember Me" checkbox.

2. style.css

Purpose: Styles the HTML page to ensure a modern and user-friendly interface.

Key Functionalities:

- Defines layout, colors, and typography for the login and registration forms.

3. script.js

Purpose: Manages client-side JavaScript for AJAX calls.

Key Functionalities:

- Handles form submissions for login and registration.
- Sends requests to the API and processes responses.

4. api/index.php

Purpose: The main API endpoint that handles routing and requests.

Key Functionalities:

- Routes requests to appropriate methods based on the action.

Functions:

```
// Entry point for API requests.
```

Parameters:

- None directly (uses global \$_SERVER data).

Return Type:

- JSON responses indicating success or failure.
-

5. api/UserService.php

Purpose: Handles user-related logic such as authentication and registration.

Key Functionalities:

- Validates user credentials.
- Manages user registration.

Functions:

authenticate(\$email, \$password)

- **Parameters:**
 - string \$email: The user's email address.
 - string \$password: The user's password.
 - **Return Type:**
 - array: Contains the status and message regarding authentication.
 - **Description:** Validates user credentials and returns a success or failure message.
-

6. api/db.php

Purpose: Establishes a connection to the MySQL database.

Key Functionalities:

- Connects to the database using PDO.

Functions:

```
// Database connection setup.
```

Parameters:

- None (uses hardcoded credentials).

Return Type:

- PDO: A PDO instance for database interaction.
-

7. api/routes.php

Purpose: Defines the API routes.

Key Functionalities:

- Maps route names to their corresponding actions.

Functions:

```
// Route definitions for the API.
```

Parameters:

- None.

Return Type:

- array: An associative array mapping route names to actions.
-

Usage Instructions

1. **Create the database:**
 - Name: login_system
 - Run SQL script from db/users.sql.
2. **Update database credentials** in db.php if necessary.
3. **Start the server** and navigate to index.html.

Conclusion

This documentation provides a comprehensive overview of the Login System project. Each component is designed to work together to facilitate user authentication and management while ensuring a clean and modern user interface.