Absolutely, victor. Let's start by setting up a basic configuration file (`config.php`) for your project. This file will contain the necessary database connection details and other common settings that your application will use. We'll then include this configuration file in your main entry point (`index.php`) and other scripts as needed.

Here's how we can set this up:

### Step 1: Create `config.php`

Create a file named `config.php` in the root directory of your project. This file will contain your database connection details and other configurations.

```php

<?php

// Database configuration

define('DB\_SERVER', 'localhost');

define('DB\_USERNAME', 'root');

define('DB\_PASSWORD', '');

define('DB\_NAME', 'rate\_clearance\_system');

// Attempt to connect to MySQL database

$conn = new mysqli(DB\_SERVER, DB\_USERNAME, DB\_PASSWORD, DB\_NAME);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Session settings

session\_start();

// Other configuration settings can be added here

?>

```

### Step 2: Update `index.php`

Create or update your `index.php` to include the configuration file. This file will handle routing and session initialization.

```php

<?php

// Include the configuration file

require\_once 'config.php';

// Determine the requested page

$requestedPage = isset($\_GET['page']) ? $\_GET['page'] : 'home';

// Route the request to the appropriate page

switch ($requestedPage) {

case 'home':

include 'home.php';

break;

case 'register':

include 'register.php';

break;

case 'login':

include 'login.php';

break;

case 'admin\_dashboard':

include 'admin\_dashboard.php';

break;

case 'finance\_dashboard':

include 'finance\_dashboard.php';

break;

case 'conveyancer\_dashboard':

include 'conveyancer\_dashboard.php';

break;

default:

include '404.php'; // Page not found

break;

}

?>

```

### Step 3: Create the Home Page (`home.php`)

Create a basic home page for your application.

```php

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Home</title>

</head>

<body>

<h1>Welcome to the Rate Clearance System</h1>

<nav>

<a href="index.php?page=register">Register</a>

<a href="index.php?page=login">Login</a>

</nav>

</body>

</html>

```

### Step 4: Update Registration and Login Forms

Make sure your registration (`register.php`) and login (`login.php`) forms include the configuration file and handle form submissions correctly.

\*\*register.php\*\*

```php

<?php

require\_once 'config.php';

// Initialize error message

$error\_message = "";

// Handle registration form submission

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['register'])) {

// Registration logic here

// ...

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>User Registration</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">

</head>

<body>

<main>

<h2>User Registration</h2>

<form method="POST" action="">

<!-- Registration form fields -->

<button type="submit" name="register">Register</button>

</form>

<?php

if (!empty($error\_message)) {

echo '<p style="color: red;">' . $error\_message . '</p>';

}

?>

</main>

</body>

</html>

```

\*\*login.php\*\*

```php

<?php

require\_once 'config.php';

// Handle login form submission

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['signin'])) {

// Login logic here

// ...

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>User Login</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">

</head>

<body>

<main>

<h2>User Login</h2>

<form method="POST" action="">

<!-- Login form fields -->

<button type="submit" name="signin">Sign In</button>

</form>

</main>

</body>

</html>

```

### Step 5: Create Dashboard Pages

Create placeholder pages for each role's dashboard, such as `admin\_dashboard.php`, `finance\_dashboard.php`, and `conveyancer\_dashboard.php`.

\*\*admin\_dashboard.php\*\*

```php

<?php

require\_once 'config.php';

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Admin Dashboard</title>

</head>

<body>

<h1>Welcome to the Admin Dashboard</h1>

</body>

</html>

```

\*\*finance\_dashboard.php\*\*

```php

<?php

require\_once 'config.php';

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Finance Director Dashboard</title>

</head>

<body>

<h1>Welcome to the Finance Director Dashboard</h1>

</body>

</html>

```

\*\*conveyancer\_dashboard.php\*\*

```php

<?php

require\_once 'config.php';

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Conveyancer Dashboard</title>

</head>

<body>

<h1>Welcome to the Conveyancer Dashboard</h1>

</body>

</html>

```

With these steps, you should have a basic configuration for your project that handles registration, login, and redirection to role-specific dashboards. If you need any further assistance or adjustments, feel free to let me know!