Developer's Guide

URL Rewrite Configuration:

To enable URL rewriting in your server configuration, follow these steps based on your web server:

Apache HTTP Server:

1. Enable the mod\_rewrite module in Apache.

2. Update the Apache configuration file (e.g., httpd.conf) to allow .htaccess file usage.

3. Uncomment the line "LoadModule rewrite\_module modules/mod\_rewrite.so" to enable the module.

4. Find the <Directory> section for your website and set "AllowOverride All" to enable .htaccess file usage.

5. Save the configuration file and restart Apache.

Nginx:

1. Edit the Nginx configuration file for your website (e.g., /etc/nginx/sites-available/<your\_site>).

2. Inside the server block, add a location block to handle URL rewriting.

3. For example:

```

location / {

try\_files $uri $uri/ /index.php?$query\_string;

}

```

4. Save the configuration file and restart Nginx.

Microsoft IIS:

1. Open Internet Information Services (IIS) Manager.

2. Select your website or application from the left-hand side.

3. Double-click on the "URL Rewrite" feature.

4. Click "Add Rule(s)" on the right-hand side.

5. Choose the desired rule template or create a custom rule.

6. Configure the rule settings according to your requirements.

7. Save the rule and exit the URL Rewrite feature.

LiteSpeed Web Server:

1. Access the LiteSpeed WebAdmin Console.

2. Navigate to the Configuration menu and select "Server".

3. Enable the "Rewrite Engine" in the server settings.

4. Save the configuration changes.

Caddy Server:

1. Open the Caddyfile for your website configuration.

2. Add a rewrite directive to handle URL rewriting.

3. For example:

```

rewrite {

to {path} {path}/ /index.php?{query}

}

```

4. Save the Caddyfile and restart Caddy.

Website Setup:

1. After enabling URL rewriting, you can proceed with running the website.

2. Make sure to use the shortcut "ctrl + click: user::project\_root();" to define the project root path.

3. Define the path from the server to the project folder.

- For example, if the project is located inside the root directory of localhost as "/Project/", adjust the path accordingly.

4. Ensure that the requested URI is properly handled by the website.

5. Classes are used to classify conflicting functions and to improve code manageability.

6. The web application does not follow the MVC pattern but uses a directory structure and file naming for organization and clarity.

7. Code is structured into functions for reusability and to maintain a clean and neat codebase.

8. Several external libraries and assets are used, including Bootstrap, Owl Carousel, Easing, Tempusdominus bs4, Waypoints, and Dark Reader Chrome Extension API.

- All assets are placed in the "/app/view/assets/" directory.

- Source files are located in the "/app/view/src/" directory.

9. Admin-related functions are placed in "/app/controller/adminController.php".

10. User-related and miscellaneous functions are inside "/app/controller/userController.php".

11. Lawyer functions can be found in "/app/controller/lawyerController.php".

12. Database configuration is set up in "/app/model/db.php".

- Warning: The project uses a single-page entry point and includes files based on the request by rewriting all requests to the entry point. This approach required additional time to implement, resulting in other aspects being less optimized.

13. Admin, user, and lawyer functions are organized within their respective controller files.

14. The code is designed to be easy to understand for individuals with programming knowledge.

Thank you for your attention and contribution to the project!