I am planning to initiate a project that will incorporate a meticulously crafted architecture, emphasizing cleanliness, security, and seamless connectivity with databases as fundamental pillars of its development framework.

The UserController class in a Spring Boot application serves as a RESTful controller responsible for handling user authentication-related HTTP requests. Let's break down its functionality:

REST Endpoint Mapping:

The controller is mapped to the base path "/api/v1/user/auth", indicating that all endpoints handled by this controller will start with this base path.

Dependency Injection:

It injects an instance of the UserService interface using the @Autowired annotation, indicating that the controller relies on the functionality provided by the UserService.

Register Endpoint:

The register() method is mapped to the HTTP POST request at "/register". It expects a JSON payload of type RegisterRequestDTO containing user registration data.

Upon receiving a registration request, it logs the request details and delegates the registration process to the UserService. It then returns an HTTP response containing the result of the registration process encapsulated within an AuthenticationResponseDTO.

Login Endpoint:

The login() method is mapped to the HTTP POST request at "/login". Similar to the registration endpoint, it expects a JSON payload of type LoginRequestDTO containing user login credentials.

Upon receiving a login request, it logs the request details, delegates the login process to the UserService, and returns an HTTP response containing the authentication result encapsulated within an AuthenticationResponseDTO.

Get All Users Endpoint:

The getAllUsers() method is mapped to the HTTP GET request at "/users". This endpoint retrieves a list of all users.

Upon receiving the request, it delegates the task of fetching all users to the UserService and returns an HTTP response containing the list of users.

Logging:

Throughout the controller, there are logging statements that provide visibility into the incoming requests and outgoing responses, aiding in debugging and monitoring the application.

Overall, the UserController serves as the entry point for handling user authentication-related operations, effectively routing incoming requests to appropriate methods in the UserService and returning the corresponding HTTP responses. It adheres to RESTful principles by mapping HTTP methods to specific endpoints and follows the dependency injection pattern for managing dependencies.