## High-Level Design Document:

**1. Introduction**

This document outlines the high-level design for a full-stack web application designed to showcase criminology research journals produced by students. The application will leverage a modern architecture, utilizing React for the frontend, Django (Python) for the backend, and MySQL for data persistence.

**2. System Architecture**

The application will follow a three-tier architecture:

* **Presentation Tier (Frontend):** React will be used to create a dynamic and interactive user interface. It will handle user interactions, display journal information, and communicate with the backend via API calls.
* **Application Tier (Backend):** Django will serve as the backend framework. It will handle routing, user authentication, business logic, data access, and communication with the frontend.
* **Data Tier (Database):** MySQL will store the application's data, including user information, journal details, and other relevant data.

**3. Backend Design (Django)**

**3.1. Core Components:**

* **Models:** Django models will define the database schema for storing information about users (students, faculty, admins), journals (title, author, abstract, content, publication date, etc.), and potentially other related entities (e.g., categories, keywords).
* **Views:** Django views will handle incoming requests from the front end. They will interact with the models to retrieve and manipulate data and then return responses in a format that the frontend can understand (e.g., JSON). RESTful APIs will be the primary communication method.
* **Serializers:** Django serializers will convert data between Python objects and JSON format, facilitating seamless communication between the backend and frontend.
* **URLs:** Django's URL routing system will map incoming requests to the appropriate views.
* **Authentication:** Django's built-in authentication system will manage user accounts, login/logout functionality, and access control. Different user roles (student, faculty, admin) will have different permissions.
* **Admin Panel:** Django's admin panel will provide a user-friendly interface for managing the application's data.

**3.2. Database Design (MySQL):**

The database will consist of tables corresponding to the Django models. Key tables include:

* users: Stores user information (username, password, email, role, etc.).
* journals: Stores journal details (title, author, abstract, content, publication date, etc.).
* (Potentially other tables for categories, keywords, etc.)

Relationships between tables will be defined (e.g., a journal will have a foreign key referencing the users table to indicate the author).

**4. Frontend Design (React)**

* **Components:** React components will be used to build the user interface, including components for displaying journal lists, individual journal pages, user profiles, and administrative dashboards.
* **Routing:** React Router will handle client-side routing, allowing users to navigate between different sections of the application.
* **State Management:** A state management library like Redux or Context API will be used to manage the application's state and ensure data consistency.
* **API Calls:** React components will make API calls to the Django backend to fetch and submit data.

**5. Technology Stack**

* **Frontend:** React, JavaScript, HTML, CSS
* **Backend:** Django (Python), Django REST Framework
* **Database:** MySQL
* **Hosting:** (To be determined)

**6. Deployment**

(Details on deployment strategy, including server infrastructure, CI/CD pipelines, etc., will be added in later stages.)

**7. Security Considerations**

* **Authentication and Authorization:** Secure user authentication and authorization mechanisms will be implemented to protect sensitive data.
* **Data Validation:** Input validation will be performed on both the frontend and backend to prevent vulnerabilities such as SQL injection and cross-site scripting (XSS).
* **HTTPS:** All communication between the client and server will be over HTTPS to ensure data encryption.

**8. Future Enhancements**

* Search functionality for journals.
* Commenting and rating system for journals.
* Integration with other academic resources.
* Improved user interface and user experience.