Execution Steps for College Lost & Found Management System

1. System Requirements

- Operating System: Windows / Linux / macOS
- Node.js (v16 or above)
- MongoDB (Local or MongoDB Atlas)
- Web Browser (Google Chrome recommended)
- Code Editor (VS Code)

2. Project Setup

1. Download or clone the project repository from GitHub.
2. Extract the project folder if downloaded as ZIP.
3. Open the project folder in Visual Studio Code.

3. Backend Execution Steps

1. Navigate to the backend directory:
2. cd backend
3. Install all required dependencies:
4. npm install
5. Create a .env file inside the backend folder with the following details:

   MONGO_URI=your_mongodb_connection_string
   JWT_SECRET=your_secret_key
   EMAIL_USER=your_gmail_id
   EMAIL_PASS=your_app_password
6. Start the backend server:

   npm start
7. If the setup is correct, the server will start on:

   http://localhost:5000
8. MongoDB connection confirmation message will appear in the terminal.

4. Frontend Execution Steps

1. Navigate to the frontend directory:
2. cd frontend
3. Open index.html in a web browser.
4. The homepage of the Lost & Found System will be displayed.

5. Application Usage Flow

1. User Registration
   - Student or staff registers using name, email, PRN, password, and role.
2. Login
   - User logs in using registered credentials.
   - JWT token is generated and stored for secure access.
3. Report Lost Item
   - Student reports a lost item by entering category, description, keywords, location, date, and image.
   - Data is stored in the MongoDB LostItem collection.
4. Report Found Item

    - Staff reports a found item using similar details.
    - The system automatically runs the matching algorithm.
5. Auto-Matching
    - Category, keywords, and location are compared.
    - A match score is calculated and stored.
6. Admin Verification
    - Admin logs into the admin panel.
    - Views matched items with images and match score.
    - Approves return after verification.
7. Email Notification
    - Upon approval, an automated email is sent to the student.
    - Database records are updated accordingly.
8. History Tracking
    - Users can view the status of their reported items (Pending / Matched / Returned).

6. Termination

- Stop backend server using:
- Ctrl + C
- Close browser to stop frontend usage.

7. Conclusion

The project executes successfully by integrating frontend, backend, database, image handling, authentication, and email services. It demonstrates real-time database operations, secure access control, and automated decision-making.