**This API endpoint handles user login by validating the email and password provided by the client. Upon successful login, it returns a JSON Web Token (JWT) to be used for authentication in subsequent requests.**

**Endpoint**

POST /login

**Request**

**Headers**

Content-Type: application/json

**Body**

The request body must contain the following fields in JSON format:

| Field | Type | Description |
| --- | --- | --- |
| email | String | The user’s email address |
| password | String | The user’s password |

**Response**

**Success Response**

HTTP Status Code: 200 OK

Response Body:

{

"message": "Login successful",

"token": "<JWT\_TOKEN>"

}

| Field | Type | Description |
| --- | --- | --- |
| message | String | A success message |
| token | String | A JWT token to be used for authentication in future requests |

**Error Responses**

**Invalid Email or Password**

HTTP Status Code: 400 Bad Request

Response Body:

{

"error": "Invalid email or password."

}

**General Error**

HTTP Status Code: 400 Bad Request

Response Body:

{

"error": "<Error message>"

}

The error message will vary depending on the issue encountered.

**Authentication Process**

**Database Query:**

The system queries the database to find a user with the provided email.

If no user is found, an error is returned.

**Password Validation:**

The provided password is validated using bcrypt. The stored password hash is salted and hashed with the user's email for additional security.

**JWT Generation:**

Upon successful validation, a JWT token is generated using the secret key (JWT\_SECRET) defined in the environment variables.

The token is set to expire after 1 hour.

Environment Variables

The following environment variables must be configured for the API to work:

**Database Schema**

The User table in the database should have the following fields:

| Column | Type | Description |
| --- | --- | --- |
| Email | String | The users email (primary key) |
| Password | String | The user’s hashed and salted password |