API Documentation: Authentication (updated)

This document provides the technical specification for the API endpoints related to user authentication, session management, and password reset, as defined in auth.py.

1. User Login

Authenticates a user and provides a session token.

- **Endpoint**: POST /auth/login
- **Description**: Validates a user's username and password. If the credentials are correct, it generates and returns a JWT access token containing the user's employee_id and role.
- **Authentication**: None required.

Request Body

Field	Type	Required	Description
username	String	Yes	The user's unique username
password	String	Yes	The user's plain-text password.

Example Request

```
{
        "username": "a_valid_user",
        "password": "their_correct_password"
}
```

Responses

- **200 OK (Success)**
    - **Description**: Returned when the credentials are valid.
    - **Body**:

```
{
        "access_token": "..."
}
```

- 400 Bad Request (Invalid Input)
  - **Description**: Returned if the request body is not valid JSON, if required fields are missing, or if the input is malformed.
  - **Body**:

```
{
        "message": "Invalid input",
        "errors": {
                "username": [
                        "Missing data for required field."
                ]
        }
}
```

- **401 Unauthorized (Invalid Credentials)**
  - **Description**: Returned if the username does not exist or the password is incorrect.
  - **Body**:

```
{
        “message”: “Invalid credentials”
}
```

2. User Logout

Invalidates the current user's access token to log them out.

- **Endpoint**: DELETE /auth/logout

- **Description**: Logs the user out by adding their current access token's unique identifier (jti) to a server-side blocklist. This prevents the token from being used for any further authenticated requests.
- **Authentication**: **Access Token Required**. The request must include a valid access token in the Authorization header.
- **Request Body**: (None)

Responses

- **200 OK (Success)**
    - **Description**: Returned when the token is successfully added to the blocklist.
    - **Body**:

```
{
    "message": "Successfully logged out!"
}
```

- **401 Unauthorized (Invalid Token)**
    - **Description**: Returned if the access token is missing, invalid, or expired.
    - **Body**:

```
{
    "message": "Token is invalid or expired",
    "error": "Token has expired"
}
```

3. Request Password Reset

Initiates the password reset process by sending a secure link to the user's registered email.

- **Endpoint**: POST /auth/request_reset
- **Description**: Sends an email containing a unique, time-sensitive token that the user can use to reset their password.

- **Authentication**: None required.

Request Body

Field	Type	Required	Description
email	String	Yes	The email address of the user's account.

Example Request

```
{
	"email": user@example.com
}
```

Responses

- **200 OK (Success)**
  - **Description**: Returned when the email is found and the reset link has been sent.
  - **Body**:

```
{
	"message": "Code has been sent"
}
```

- **400 Bad Request (Invalid Input)**
  - **Description**: Returned if the email field is missing from the request body.
  - **Body**:

```
{
	"message": "Missing email field in JSON"
}
```

- **404 Not Found**
  - **Description**: Returned if the provided email does not exist in the system.
  - **Body**:

```
{
        "message": "Email not found"
}
```

4. Reset Password

Sets a new password for a user account using a valid reset token.

- **Endpoint**: PUT /auth/reset
- **Description**: Verifies the password reset token and updates the user's account with the new password. The token is valid for **5 minutes** (300 seconds).
- **Authentication**: None required.

Query Parameters

Parameter	Type	Required	Description
token	String	Yes	The secure token received in the password reset email.

Request Body

Field	Type	Required	Description
new_password	String	Yes	The user's new plain-text password.

Example Request URL

```
{{BaseURL}}/auth/reset?token=<SECURE_TOKEN>
```

Responses

- **200 OK (Success)**
    - **Description**: Returned when the token is valid and the password has been successfully updated.
    - **Body**:

```
{
    "message": "Password reset successfully"
}
```

- **400 Bad Request (Invalid Input)**
    - **Description**: Returned if the token is missing from the query parameters or if the new_password is missing from the request body.
    - **Body**:

```
{
    "message": "Missing new password"
}
```

- **500 Internal Server Error (Invalid Token)**
    - **Description**: An "Unexpected error occurred" will be returned if the token is invalid or expired, as this raises an exception during deserialization.
    - **Body**:

```
{
    "message": "Unexpected error occurred",
    "error": "Signature expired"
}
```