User Authentication API Documentation

Document Overview

Document Title: User Authentication API Documentation
Author: Priyadharshini N S
Version: 1.0
Last Updated: November 2025
Purpose: This document provides information about the User Authentication API endpoints, including request methods, parameters, and response examples.
Audience: Developers and testers integrating authentication features into web or mobile applications.

---

1. Introduction

The **User Authentication API** enables secure login, registration, and token management for applications that require user-level access control.
This document outlines all available endpoints, request and response structures, and sample implementations.

---

2. Base URL

```
https://api.projectx.com/v1
```

All endpoints in this document use the above base URL.

---

3. Authentication Method

The API uses **Bearer Token Authentication**.
Each request (except registration and login) must include an authorization header:

```
Authorization: Bearer <access_token>
```

Tokens are generated after a successful login.

---

4. Endpoints Overview

Endpoint	Method	Description
`/auth/register`	POST	Registers a new user account
`/auth/login`	POST	Authenticates user credentials and returns a token
`/auth/user`	GET	Retrieves user profile details
`/auth/refresh`	POST	Refreshes expired access tokens

---

5. Endpoints Details

5.1 Register User

Endpoint:

```
POST /auth/register
```

Request Body:

```
{
  "name": "Priya",
  "email": "priya@example.com",
  "password": "StrongPassword@123"
}
```

Response:

```
{
  "status": "success",
  "message": "User registered successfully"
}
```

Response Code: `201 Created`

5.2 Login User

Endpoint:

```
POST /auth/login
```

Request Body:

```
{
  "email": "priya@example.com",
  "password": "StrongPassword@123"
}
```

Response:

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIs...",
  "refresh_token": "eyJhbGciOiJIUzI1NiIs...",
  "token_type": "Bearer",
  "expires_in": 3600
}
```

Response Code: `200 OK`

5.3 Get User Details

Endpoint:

```
GET /auth/user
```

Headers:

```
Authorization: Bearer <access_token>
```

Response:

```
{
  "id": "001",
  "name": "Priya",
  "email": "priya@example.com",
  "role": "user"
}
```

Response Code: `200 OK`

5.4 Refresh Token

Endpoint:

```
POST /auth/refresh
```

Request Body:

```
{
  "refresh_token": "eyJhbGciOiJIUzI1NiIs..."
}
```

Response:

```
{
  "access_token": "new-access-token",
  "expires_in": 3600
}
```

Response Code: `200 OK`

---

6. Error Codes

Error Code	Description	Possible Cause

400	Bad Request	Missing or invalid fields
401	Unauthorized	Invalid or expired token
403	Forbidden	Access denied for this user
404	Not Found	Endpoint not available
500	Internal Server Error	Unexpected server issue

7. Sample cURL Commands

Login Example

```
curl -X POST https://api.projectx.com/v1/auth/login \
-H "Content-Type: application/json" \
-d '{"email":"priya@example.com","password":"StrongPassword@123"}'
```

Get User Details Example

```
curl -X GET https://api.projectx.com/v1/auth/user \
-H "Authorization: Bearer <access_token>"
```

8. Version History

Version	Date	Description	Author
1.0	November 2025	Initial Release	Priyadharshini N S

9. Support

For API support or questions, contact:
📧 **api-support@projectx.com**
🌐 https://example.com/api/docs