# **API Documentation**

This document details all API endpoints for the project, including their base URLs, methods, and sample responses.

# **1. User Service**

### **Base URL**

https://blog-platform-bdn4.vercel.app/api/

### **Endpoints**

## **1.1 Register User**

* **URL**: /register/
* **Method**: POST
* **Description**: Registers a new user.

**Request Body**:  
 {

"username": "string",

"email": "string",

"password": "string"

}

**Response**:  
 {

"id": 1,

"username": "string",

"email": "string"

}

#### **1.2 Login**

* **URL**: /login/
* **Method**: POST
* **Description**: Logs in a user and provides an authentication token.

**Request Body**:  
 {

"username": "string",

"password": "string"

}

**Response**:  
 {

"token": "string"

}

#### **1.3 User Details**

* **URL**: /users/<id>/
* **Methods**:
  1. **GET**: Retrieve details of a specific user.

**Response**:  
 {

"id": 1,

"username": "string",

"email": "string"

}

* 1. **PUT**: Update user details (authenticated users or admin only).

**Request Body**:  
 {

"username": "new\_username",

"email": "new\_email@example.com"

}

**Headers**:  
 Authorization: Bearer <token>

**Response**:  
 {

"id": 1,

"username": "new\_username",

"email": "new\_email@example.com"

}

* 1. **DELETE**: Delete a user account (authenticated users or admin only).

**Headers**:  
 Authorization: Bearer <token>

**Response**:  
 {

"detail": "User deleted successfully."

}

# **2. Blog Service**

### **Base URL**

https://blog-platform-bupu.vercel.app/api/

### **Endpoints**

## **2.1 Create Blog Post**

* **URL**: /blogs/
* **Method**: POST
* **Description**: Creates a new blog post.

**Request Body**:  
 {

"title": "string",

"content": "string"

}

**Headers**:  
 Authorization: Bearer <token>

**Response**:  
 {

"id": 1,

"title": "string",

"content": "string",

"author\_id": 1,

"created\_at": "datetime",

"updated\_at": "datetime"

}

#### **2.2 List Blog Posts**

* **URL**: /blogs/
* **Method**: GET
* **Description**: Retrieves a list of all blog posts.

**Response**:  
 [

{

"id": 1,

"title": "string",

"content": "string",

"author\_id": 1,

"created\_at": "datetime",

"updated\_at": "datetime"

}

]

#### **2.3 Blog Post Details**

* **URL**: /blogs/<id>/
* **Methods**:
  1. **GET**: Retrieve details of a specific blog post.

**Response**:  
 {

"id": 1,

"title": "string",

"content": "string",

"author\_id": 1,

"created\_at": "datetime",

"updated\_at": "datetime"

}

* 1. **PUT**: Update a specific blog post (authenticated users or admin only).

**Request Body**:  
 {

"title": "updated title",

"content": "updated content"

}

**Headers**:  
 Authorization: Bearer <token>

**Response**:  
 {

"id": 1,

"title": "updated title",

"content": "updated content",

"author\_id": 1,

"created\_at": "datetime",

"updated\_at": "datetime"

}

* 1. **DELETE**: Delete a specific blog post (authenticated users or admin only).

**Headers**:  
 Authorization: Bearer <token>

**Response**:  
 {

"detail": "Blog post deleted successfully."

}

# **3. Comment Service**

### **Base URL**

https://commetervice.vercel.app/api/

### **Endpoints**

## **3.1 Add Comment**

* **URL**: /comments/
* **Method**: POST
* **Description**: Adds a comment to a blog post.

**Request Body**:  
 {

"post\_id": 1,

"content": "string"

}

**Headers**:  
 Authorization: Bearer <token>

**Response**:  
 {

"id": 1,

"blog\_post\_id": 1,

"user\_id": 1,

"content": "string",

"created\_at": "datetime"

}

#### **3.2 List Comments for a Blog Post**

* **URL**: /comments/?post\_id=<id>
* **Method**: GET
* **Description**: Retrieves all comments for a specific blog post.

**Response**:  
 [

{

"id": 1,

"blog\_post\_id": 1,

"user\_id": 1,

"content": "string",

"created\_at": "datetime"

}

]

### **Authorization**

All POST, PUT, and DELETE endpoints require authentication via a Bearer token:  
 Authorization: Bearer <your-token>

**Decisions made and Trade-offs:**

The assignment required **PUT /users/<id>** to edit a blog post and **DELETE /users/<id>** to delete a blog post. However, I implemented these endpoints for user-related operations: **PUT /users/<id>** edits a user profile, and **DELETE /users/<id>** deletes a user account. This decision prioritizes API clarity and aligns endpoints with their intended resource, following REST principles. The tradeoff is deviating from the assignment's instructions but achieving better long-term maintainability and understanding.