**API Documentation**

1. **Authentication API :**

**Endpoints:**

**POST /api/v1/auth/signup :**

**Description:** Creates a new user account.

**Request Body:**

{

"name": "John Doe",

"email": "john@example.com",

"password": "password123"

}

**Response:**

{

"status": "success",

"token": "JWT\_TOKEN",

"data": {

"user": {

"name": "John Doe",

"email": "john@example.com"

}

}

}

**POST /api/v1/auth/login:**

**Description:** Logs in an existing user.

**Request Body:**

{

"email": "john@example.com",

"password": "password123"

}

**Response:**

{

"status": "success",

"token": "JWT\_TOKEN",

"data": {

"user": {

"name": "John Doe",

"email": "john@example.com"

}

}

}

**Protected Routes :**

**Middleware:**

**protect:** Middleware to protect routes by verifying JWT token.

**restrictTo:** Middleware to restrict access based on user roles.

**Usage Guidelines:**

**Signup:**

Users should provide a name, email, and password in the request body.

Upon successful signup, a JWT token is generated and sent in the response along with user data.

**Login:**

Users should provide their email and password in the request body.

Upon successful login, a JWT token is generated and sent in the response along with user data.

**Protect Middleware:**

Protects routes by verifying the JWT token in the request headers.

Grants access to the protected route if the token is valid and the user exists.

**RestrictTo Middleware:**

Restricts access to routes based on user roles specified as arguments.

Only users with specified roles are granted access to the route.

1. **Cart API :**

**Endpoints:**

**GET /api/v1/cart/items:**

**Description:** Retrieves items in the user's cart.

**Request Headers:**

**Authorization:** Bearer JWT\_TOKEN

**Response:**

{

"status": "success",

"data": [

{

"\_id": "60983d60242cc9a42c7c823d",

"product": {

"\_id": "60983035efb47391c3b11846",

"name": "Product Name",

"price": 100,

"quantity": 5

},

"quantity": 2,

"price": 200

},

...

]

}

**POST /api/v1/cart/items/:id :**

**Description:** Adds a product to the user's cart.

**Request Headers:**

**Authorization:** Bearer JWT\_TOKEN

**Request Parameters:**

**id:** Product ID

**Response:**

{

"status": "success",

"numOfCartItems": 3,

"data": {

"\_id": "60985a0f02e9b4e3a0b94b6a",

"userId": "60983035efb47391c3b11846",

"cartItems": [

{

"\_id": "60985a0f02e9b4e3a0b94b6b",

"product": "60983035efb47391c3b11846",

"price": 100,

"quantity": 1

},

...

],

"totalCartPrice": 300

}

}

**Usage Guidelines:**

**Get Items:**

Send a GET request to **/api/v1/cart/items** with the user's JWT token in the Authorization header to retrieve items in the user's cart.

The response contains an array of cart items with details such as product ID, name, price, quantity, etc.

**Add Item:**

Send a POST request to **/api/v1/cart/items/:id** with the product ID as a route parameter and the user's JWT token in the Authorization header to add a product to the user's cart.

If the product is not found or out of stock, an error response is returned.

If the product is successfully added to the cart, the response contains the updated cart data including the total price and the number of cart items.

1. **Order API :**

**Endpoints:**

**POST /api/v1/orders/create:**

**Description:** Creates a new order.

**Request Headers:**

**Authorization:** Bearer JWT\_TOKEN

**Response:**

{

"status": "success",

"data": {

"\_id": "60983d60242cc9a42c7c823d",

"user": "60983035efb47391c3b11846",

"cartItems": [

{

"product": "60983035efb47391c3b11846",

"quantity": 2,

"price": 200

},

...

],

"totalOrderPrice": 200

}

}

**GET /api/v1/orders:**

**Description:** Retrieves all orders placed by the current user.

**Request Headers:**

**Authorization:** Bearer JWT\_TOKEN

**Response:**

{

"status": "success",

"data": [

{

"\_id": "60983d60242cc9a42c7c823d",

"user": "60983035efb47391c3b11846",

"cartItems": [

{

"product": "60983035efb47391c3b11846",

"quantity": 2,

"price": 200

},

...

],

"totalOrderPrice": 200

},

...

]

}

**GET /api/v1/orders/delivered:**

**Description:** Retrieves all delivered orders placed by the current user.

**Request Headers:**

**Authorization:** Bearer JWT\_TOKEN

**Response:**

{

"status": "success",

"data": [

{

"\_id": "60983d60242cc9a42c7c823d",

"user": "60983035efb47391c3b11846",

"cartItems": [

{

"product": "60983035efb47391c3b11846",

"quantity": 2,

"price": 200

},

...

],

"totalOrderPrice": 200

},

...

]

}

**GET /api/v1/orders/undelivered:**

**Description:** Retrieves all undelivered orders placed by the current user.

**Request Headers:**

**Authorization:** Bearer JWT\_TOKEN

**Response:**

{

"status": "success",

"data": [

{

"\_id": "60983d60242cc9a42c7c823d",

"user": "60983035efb47391c3b11846",

"cartItems": [

{

"product": "60983035efb47391c3b11846",

"quantity": 2,

"price": 200

},

...

],

"totalOrderPrice": 200

},

...

]

}

**Usage Guidelines:**

**Create Order:**

Send a POST request to **/api/v1/orders/create** with the user's JWT token in the Authorization header to create a new order.

The response contains the newly created order data including cart items and total order price.

**View All Orders:**

Send a GET request to **/api/v1/orders** with the user's JWT token in the Authorization header to retrieve all orders placed by the current user.

**View Delivered Orders:**

Send a GET request to **/api/v1/orders/delivered** with the user's JWT token in the Authorization header to retrieve all delivered orders placed by the current user.

**View Undelivered Orders:**

Send a GET request to **/api/v1/orders/undelivered** with the user's JWT token in the Authorization header to retrieve all undelivered orders placed by the current user.

1. **Product API :**

**Endpoints:**

**GET /api/v1/products**

**Description:** Retrieves all products.

**Request Query Parameters:**

**price[gte]:** Filter products with a price greater than or equal to the specified value.

**price[lte]:** Filter products with a price less than or equal to the specified value.

**sort:** Sort products based on specified fields (e.g., price, createdAt).

**fields:** Select specific fields to include in the response.

**page:** Specify the page number for paginated results.

**limit:** Specify the maximum number of products per page.

**Response:**

{

"status": "success",

"data": [

{

"\_id": "60983d60242cc9a42c7c823d",

"name": "Product Name",

"description": "Product Description",

"price": 100,

...

},

...

]

}

**GET /api/v1/products/:id**

**Description:** Retrieves a specific product by ID.

**Request Parameters:**

**id:** Product ID

**Response:**

{

"status": "success",

"data": {

"\_id": "60983d60242cc9a42c7c823d",

"name": "Product Name",

"description": "Product Description",

"price": 100,

...

}

}

**POST /api/v1/products**

**Description:** Adds new product.

**Request Body :**

{

"name": "Product Name",

"description": "Product Description",

"price": 100,

...

}

**Response:**

{

"status": "success",

"data": {

"\_id": "60983d60242cc9a42c7c823d",

"name": "Product Name",

"description": "Product Description",

"price": 100,

...

}

}

**PATCH /api/v1/products/:id**

**Description:** Updates an existing product by ID.

**Request Parameters:**

**id:** Product ID

**Request Body:**

{

"name": "Updated Product Name",

"description": "Updated Product Description",

"price": 120,

...

}

**Response:**

{

"status": "success",

"data": {

"\_id": "60983d60242cc9a42c7c823d",

"name": "Updated Product Name",

"description": "Updated Product Description",

"price": 120,

...

}

}

**DELETE/api/v1/products/:id**

**Description:** Deletes an existing product by ID.

**Request Parameters:**

**id:** Product ID

**Request Body:**

{

"status": "success",

"data": null

}

**Usage Guidelines:**

**Get All Products:**

Send a GET request to **/api/v1/products** to retrieve all products.

Use query parameters to filter, sort, and paginate results.

**Get Product by ID:**

Send a GET request to **/api/v1/products/:id** to retrieve a specific product by its ID.

**Add Product:**

Send a POST request to **/api/v1/products** with the product details in the request body to add a new product.

**Update Product:**

Send a PATCH request to **/api/v1/products/:id** with the updated product details in the request body to modify an existing product.

**Delete Product:**

Send a DELETE request to **/api/v1/products/:id** to delete an existing product by its ID.

1. **Review API :**

**Endpoints:**

**GET /api/v1/products/:productId/reviews**

**Description:** Retrieves all reviews for a specific product.

**Request Parameters:**

**productId:** ID of the product for which reviews are requested.

**Response:**

{

"status": "success",

"data": [

{

"\_id": "60983d60242cc9a42c7c823d",

"product": "60983035efb47391c3b11846",

"user": "60983035efb47391c3b11847",

"rating": 4,

"comment": "Review comment text",

...

},

...

]

}

**POST /api/v1/products/:productId/reviews**

**Description:** Adds a new review for a specific product.

**Request Parameters:**

**productId:** ID of the product for which the review is being added.

**Request Body:**

{

"rating": 4,

"comment": "Review comment text",

...

}

**Response:**

{

"status": "success",

"data": {

"\_id": "60983d60242cc9a42c7c823d",

"product": "60983035efb47391c3b11846",

"user": "60983035efb47391c3b11847",

"rating": 4,

"comment": "Review comment text",

...

}

}

**PATCH /api/v1/products/:productId/reviews/:reviewId**

**Description:** Updates an existing review for a specific product.

**Request Parameters:**

**productId:** ID of the product to which the review belongs.

**reviewId:** ID of the review to be updated.

**Request Body:**

{

"rating": 5,

"comment": "Updated review comment",

...

}

**Response:**

{

"status": "success",

"data": {

"\_id": "60983d60242cc9a42c7c823d",

"product": "60983035efb47391c3b11846",

"user": "60983035efb47391c3b11847",

"rating": 5,

"comment": "Updated review comment",

...

}

}

**DELETE /api/v1/products/:productId/reviews/:reviewId**

**Description:** Deletes an existing review for a specific product.

**Request Parameters:**

**productId:** ID of the product to which the review belongs.

**reviewId:** ID of the review to be deleted.

**Response:**

{

"status": "success",

"data": null

}

**Usage Guidelines:**

**Get Reviews for Product:**

Send a GET request to **/api/v1/products/:productId/reviews** to retrieve all reviews for a specific product.

**Add Review:**

Send a POST request to **/api/v1/products/:productId/reviews** with the review details in the request body to add a new review for a specific product.

**Update Review:**

Send a PATCH request to **/api/v1/products/:productId/reviews/:reviewId** with the updated review details in the request body to modify an existing review.

**Delete Review:**

Send a DELETE request to **/api/v1/products/:productId/reviews/:reviewId** to delete an existing review for a specific product.