CINEC CAMPUS
Beyond A Graduate

BSc. (Hons) in Software Engineering
Faculty of Computing

API Documentation

Name of the Student: J. L. G. N. S. Jayawardhana
Student Registration Number: IT_IFLS_001/B003/0023
Module: IT3113
Submission Date: 06/05/2025

Base URL

https://your-domain.com/api

(Replace with your actual domain or use http://localhost:PORT/api in development)

User APIs (/users)

These endpoints allow users to register, log in, manage their cart, and update profile information.

Register User

POST /users/register

Registers a new user.

Request Body:

```
{
  "firstName": "John",
  "lastName": "Doe",
  "email": "john@example.com",
  "password": "yourPassword"
}
Responses:
- 201 Created – Success
- 400 Bad Request – Missing fields or user exists
- 500 Server Error
```

Login User

POST /users/login

Authenticates a user and sets a session cookie. Also merges guest cart.

Request Body:

```
{
  "email": "john@example.com",
  "password": "yourPassword",
  "guestCart": [
    {
      "productId": "book123",
      "title": "Book Title",
      "quantity": 1
    }
  ]
}
```

Responses:
- 200 OK – Returns user data and sets bookstore_token cookie
- 401 Unauthorized – Invalid credentials

Logout User

GET /users/logout

Logs the user out by clearing the bookstore_token cookie.

Response: 200 OK – Cookie cleared

Get Cart

GET /users/cart

Retrieves the current user's cart. Requires bookstore_token cookie.

Response:

```
{
  "cart": [
    {
      "bookId": "book123",
      "title": "Book Title",
      "quantity": 1
    }
  ]
}
```

Update Cart

POST /users/cart

Updates the user's cart with the provided item. Requires bookstore_token cookie.

Request Body:

```
{
  "bookId": "book123",
  "title": "Book Title",
  "quantity": 1
}
```

Response: Returns updated cart.

Get All Users

GET /users

Returns a list of all registered users.

Get User by ID

GET /users/:id

Returns user details for the given ID.

Update User

PUT /users/:id

Updates user information for the given ID.

Request Body:

```
{
  "firstName": "Johnny",
  "lastName": "Doel",
  "email": "johnny@example.com",
  "password": "newPassword"
}
```

Delete User

DELETE /users/:id

Deletes the user with the specified ID.

Payment API (/payment)

These endpoints integrate with Stripe to create payment sessions.

Create Stripe Checkout Session

```
POST /payment/create-payment-intent
```

Creates a Stripe Checkout session for purchasing products.

Request Body:

```
{
  "products": [
    {
      "title": "Book Title",
      "description": "Book Description",
      "price": 29.99,
      "quantity": 2,
      "image": "https://link-to-image"
    }
  ]
}
Response:
{
  "id": "stripe_session_id"
}
Redirect the user to https://checkout.stripe.com/pay/<id>
```