

API Specification Document

Course SWAP Web Application

Table of Contents

USER registration.....	2
REQUEST	2
RESPONSE	2
USER login.....	2
REQUEST	2
RESPONSE	2
SWAP requests.....	3
REQUEST.....	3
RESPONSE.....	3
GET ALL SWAP requests.....	3
RESPONSE.....	3
GET SWAP requests by user	4
RESPONSE.....	4
DELETE a SWAP request.....	4
RESPONSE.....	4
Find Matching SWAPs.....	5
RESPONSE.....	5
Confirm a SWAP.....	5
REQUEST.....	5
RESPONSE.....	5
Send Match NOTIFICATION	6
REQUEST.....	6
RESPONSE.....	6
GET NOTIFICATIONS.....	6
RESPONSE.....	6

AUTHENTICATION API

1. USER registration

- **Endpoint:** POST /api/auth/register
- **Description:** Registers a new user using their university email

REQUEST

```
{  
  "email": "student@university.edu",  
  "password": "securepassword",  
  "fullName": "John Doe"  
}
```

RESPONSE

```
{  
  "message": "User registered successfully",  
  "userId": "user123"  
}
```

2. USER login

- **Endpoint:** POST /api/auth/login
- **Description:** Logs in a user and returns an authentication token.

REQUEST

```
{  
  "email": "student@university.edu",  
  "password": "securepassword"  
}
```

RESPONSE

```
{  
  "token": "jwt-token",  
  "userId": "user123"  
}
```

COURSE SWAP API

3. Create a SWAP request

- **Endpoint:** POST /api/swaps/create
- **Description:** Allows a user to input a swap request.

REQUEST

```
{  
  "userId": "user123",  
  "offeringCourse": "CS101",  
  "desiredCourse": "CS201",  
  "semester": "Spring 2025"  
}
```

RESPONSE

```
{  
  "message": "Swap request created",  
  "swapId": "swap456"  
}
```

4. Get ALL SWAP requests

- **Endpoint:** GET /api/swaps
- **Description:** Retrieves all swap requests for browsing.

RESPONSE

```
{  
  "swapId": "swap456",  
  "userId": "user123",  
  "offeringCourse": "CS101",  
  "desiredCourse": "CS201",  
  "semester": "Spring 2025"  
}
```

5. Get SWAP requests by user

- **Endpoint:** GET /api/swaps/user/:userId
- **Description:** Fetches swap requests for a specific user.

RESPONSE

```
[
  {
    "swapId": "swap456",
    "offeringCourse": "CS101",
    "desiredCourse": "CS201",
    "semester": "Spring 2025"
  }
]
```

6. DELETE a SWAP request by user

- **Endpoint:** DELETE /api/swaps/:swapId
- **Description:** Allows a user to delete their swap request.

RESPONSE

```
{
  "message": "Swap request deleted"
}
```

MATCHING API

7. Finding Matching SWAPs

- **Endpoint:** GET /api/matches/:userId
- **Description:** Find potential swap matches for the user

RESPONSE

```
[
  {
    "matchId": "match789",
    "swapId": "swap456",
    "matchingUserId": "user456",
    "offeringCourse": "CS201",
    "desiredCourse": "CS101"
  }
]
```

8. Confirm a SWAP

- **Endpoint:** POST /api/swaps/confirm
- **Description:** Confirms a swap between two users.

REQUEST

```
{
  "user1Id": "user123",
  "user2Id": "user456",
  "swap1Id": "swap456",
  "swap2Id": "swap789"
}
```

RESPONSE

```
{
  "message": "Swap confirmed"
}
```

NOTIFICATION API

9. Send Match notifications

- **Endpoint:** POST /api/notifications/send
- **Description:** Sends a notification when a match is found

REQUEST

```
{
  "userId": "user123",
  "message": "A match has been found for your course swap request!"
}
```

RESPONSE

```
{
  "message": "Notification sent"
}
```

10. Get Notifications

- **Endpoint:** GET /api/notifications/:userId
- **Description:** Retrieves all notifications for a user.

RESPONSE

```
[
  {
    "notificationId": "notif123",
    "message": "A match has been found for your course swap request!"
  }
]
```