MIST 460 - Swagger UI - Course Recommender

Start Server

Run the FastAPI app locally (VS Code Terminal):

```
python -m uvicorn main:app --reload --host 127.0.0.1 --port 8000
```

Other Common Commands

Install dependencies:

```
pip install fastapi uvicorn
```

If you use SQLite and SQLModel / SQLAlchemy:

```
pip install sqlmodel sqlalchemy
```

To stop the server:

```
CTRL + C
```

Swagger UI

Open:

http://127.0.0.1:8000/docs

Usage Steps:

1. Click **“Try it out”**
2. Enter parameters or JSON body
3. Click **“Execute”**

Quick cURL Commands

1. Validate User ✅

ChatGPT said:

*Use **POST** for validateUser because it sends **sensitive data** (username, password) in the request body and performs an **action** (authentication), not just data retrieval.*

POST `/validate_user`

```
curl -X POST "http://127.0.0.1:8000/validate_user" \
-H "Content-Type: application/json" \
-d '{"username":"mjordan@wvu.edu","password":"0x01"}'
```

Edit Value | Schema

```
{"username":"mjordan@wvu.edu","password":"0x01"}
```

Response body


```
{
  "valid": true,
  "user": {
    "AppUserID": 1,
    "FullName": "Michael Jordan"
  }
}
```


---

2. Find Current Semester Course Offerings ✅

GET `/find_current_semester_course_offerings`

```
curl "http://127.0.0.1:8000/find_current_semester_course_offerings?subject_code=MIST&course_number=460"
```

Name	Description
subject_code * required string *(query)*	MIST
course_number * required string *(query)*	460

Response body


```
{
  "data": [
    {
      "SubjectCode": "MIST",
      "CourseNumber": "460",
      "CRN": 30002,
      "CourseOfferingSemester": "Fall",
      "CourseOfferingYear": 2025
    },
    {
      "SubjectCode": "MIST",
      "CourseNumber": "460",
      "CRN": 30003,
      "CourseOfferingSemester": "Fall",
      "CourseOfferingYear": 2025
    },
    {
      "SubjectCode": "MIST",
      "CourseNumber": "460",
      "CRN": 30004,
      "CourseOfferingSemester": "Fall",
      "CourseOfferingYear": 2025
    }
  ]
}
```


---

3. Find Prerequisites ✅

GET `/find_prerequisites`

```
curl
"http://127.0.0.1:8000/find_prerequisites?subject_code=MIST&course_number=452"
```

subject_code * required
string
(query)

MIST

course_number * required
string
(query)

460

Response body

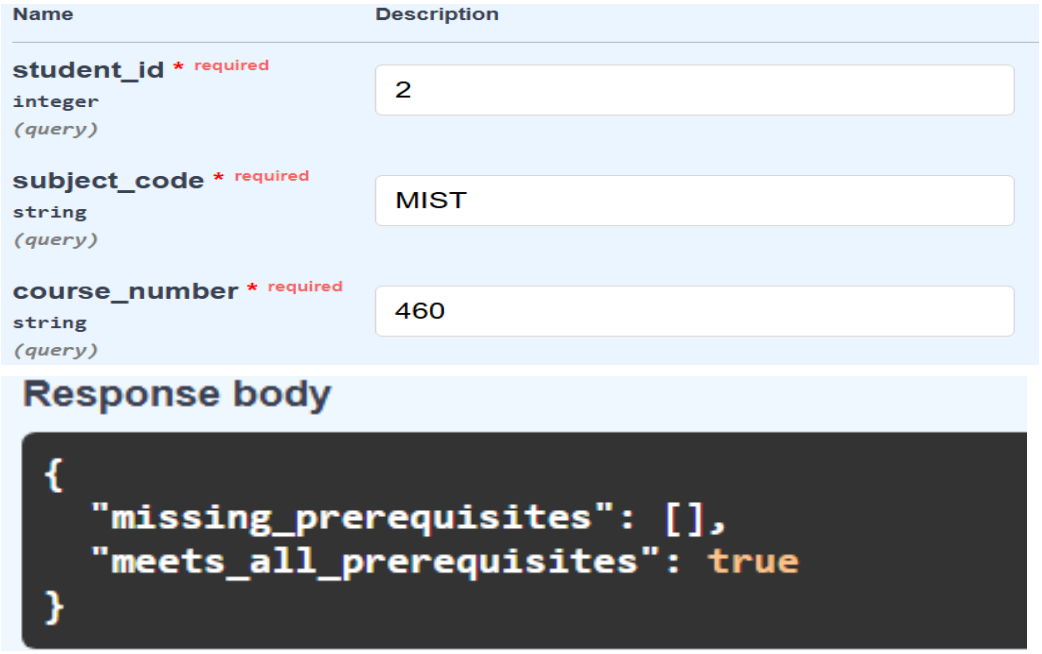

```
{
  "data": [
    {
      "SubjectCode": "MIST",
      "CourseNumber": "450"
    },
    {
      "SubjectCode": "MIST",
      "CourseNumber": "452"
    }
  ]
}
```


---

4. Check if Student Has Taken All Prerequisites ✅

GET `/check_if_student_has_taken_all_prerequisites_for_course`

```
curl "http://127.0.0.1:8000/check_if_student_has_taken_all_prerequisites_for_course?student_id=1&subject_code=MIST&course_number=460"
```

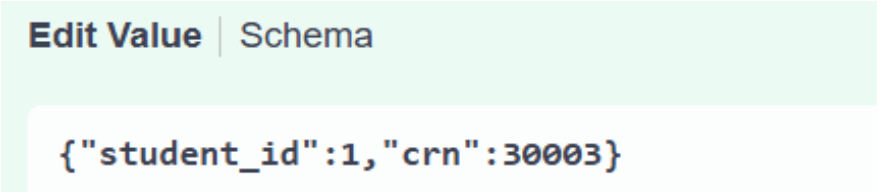


---

5. Enroll Student in Course Offering ✅

POST `/enroll_student_in_course_offering`

```
curl -X POST "http://127.0.0.1:8000/enroll_student_in_course_offering"
\
-H "Content-Type: application/json" \
-d '{"student_id":1,"crn":30003}'
```



Response body


```
{
  "success": true,
  "data": [
    {
      "EnrollmentResponse": "Enrollment successful",
      "EnrollmentSucceeded": true
    }
  ]
}
```


---

6. Get Student Enrolled Course Offerings ✅

GET `/get_student_enrolled_course_offerings`

```
curl "http://127.0.0.1:8000/get_student_enrolled_course_offerings?student_id=1"
```

student_id * required
integer
(query)

1

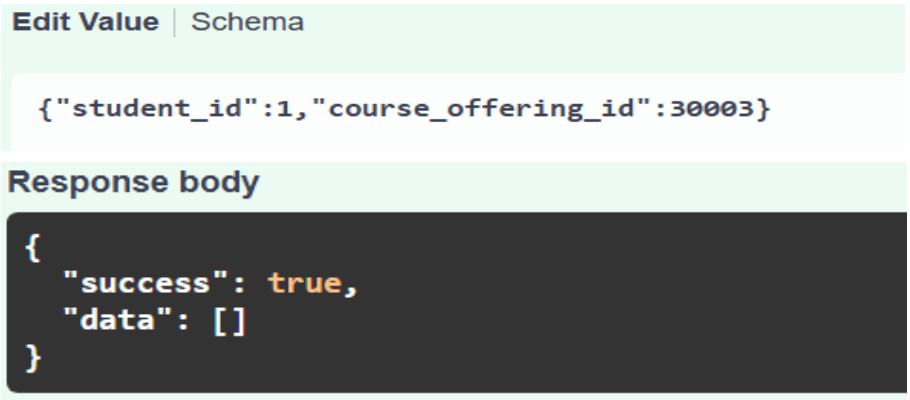
Response body


```
{
  "data": [
    {
      "RegistrationCourseOfferingID": 7,
      "EnrollmentStatus": "Completed",
      "CourseOfferingID": 3,
      "CRN": 10003,
      "CourseOfferingSemester": "Fall",
      "CourseOfferingYear": 2024,
      "SubjectCode": "MIST",
      "CourseNumber": "353",
      "Title": "Advanced Information Technology"
    },
    {
      "RegistrationCourseOfferingID": 9,
      "EnrollmentStatus": "Completed",
      "CourseOfferingID": 5,
      "CRN": 10005,
      "CourseOfferingSemester": "Fall",
      "CourseOfferingYear": 2024,
      "SubjectCode": "MIST",
      "CourseNumber": "452",
      "Title": "Systems Design/Development"
    },
    {
      "RegistrationCourseOfferingID": 22,
      "EnrollmentStatus": "Completed",
```

7. Drop Student from Course Offering ✅

POST `/drop_student_from_course_offering`

```
curl -X POST "http://127.0.0.1:8000/drop_student_from_course_offering"
\
-H "Content-Type: application/json" \
-d '{"student_id":1,"course_offering_id":30003}'
```



This means the **stored procedure didn't return any rows** from SQL Server.

Interpretation

- **Connection and SQL call succeeded**
- **No matching record was updated or selected**