Project 1

Titan Online

Group 8

Joel Anil John

Marco Gabriel

Vivian Cao

~~Maria Fernandez~~

~~Anurag Ganji~~

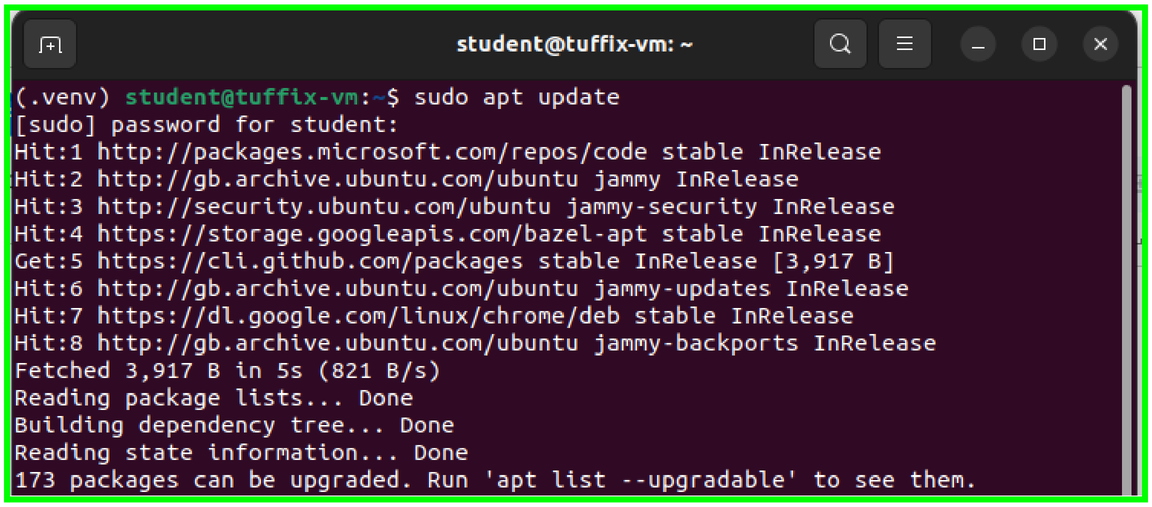
CPSC 449-01 13661

Fall Semester

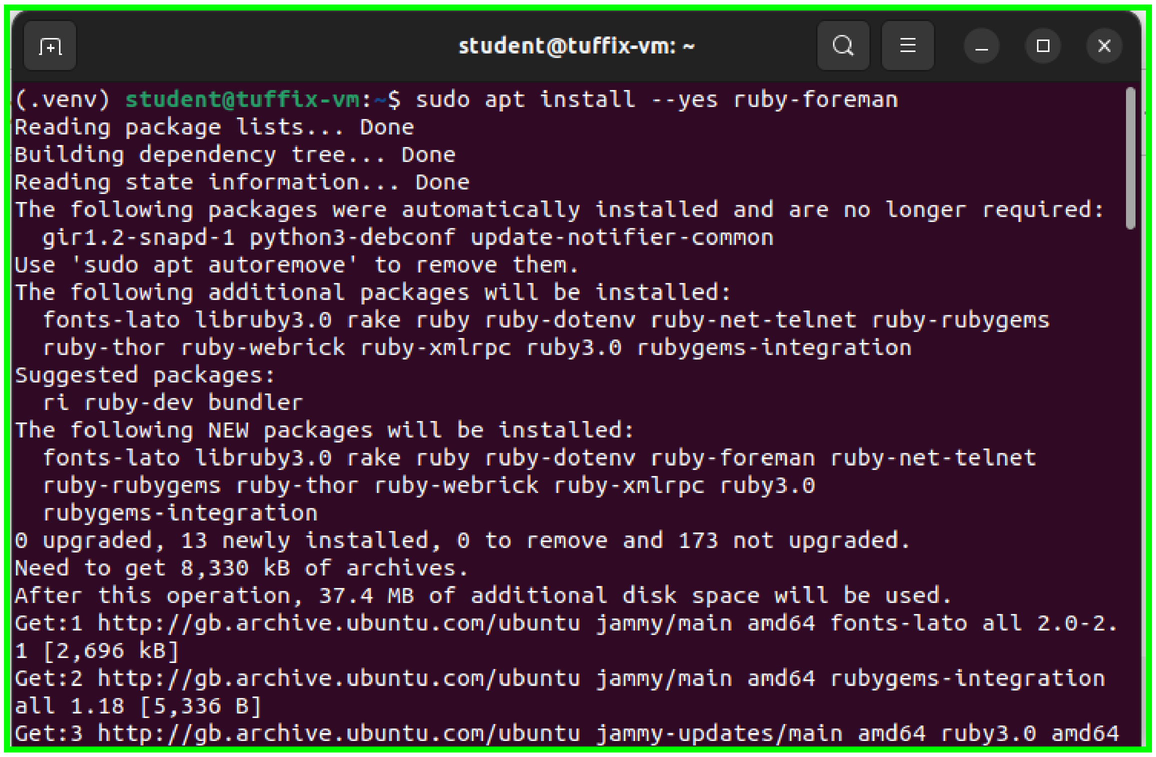
Table of Contents:

Environment Setup:

Installing Foreman:

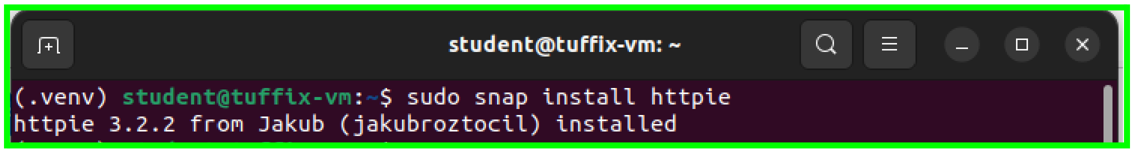


```
(.venv) student@tuffix-vm:~$ sudo apt update
[sudo] password for student:
Hit:1 http://packages.microsoft.com/repos/code stable InRelease
Hit:2 http://gb.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:4 https://storage.googleapis.com/bazel-apt stable InRelease
Get:5 https://cli.github.com/packages stable InRelease [3,917 B]
Hit:6 http://gb.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:7 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:8 http://gb.archive.ubuntu.com/ubuntu jammy-backports InRelease
Fetched 3,917 B in 5s (821 B/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
173 packages can be upgraded. Run 'apt list --upgradable' to see them.
```



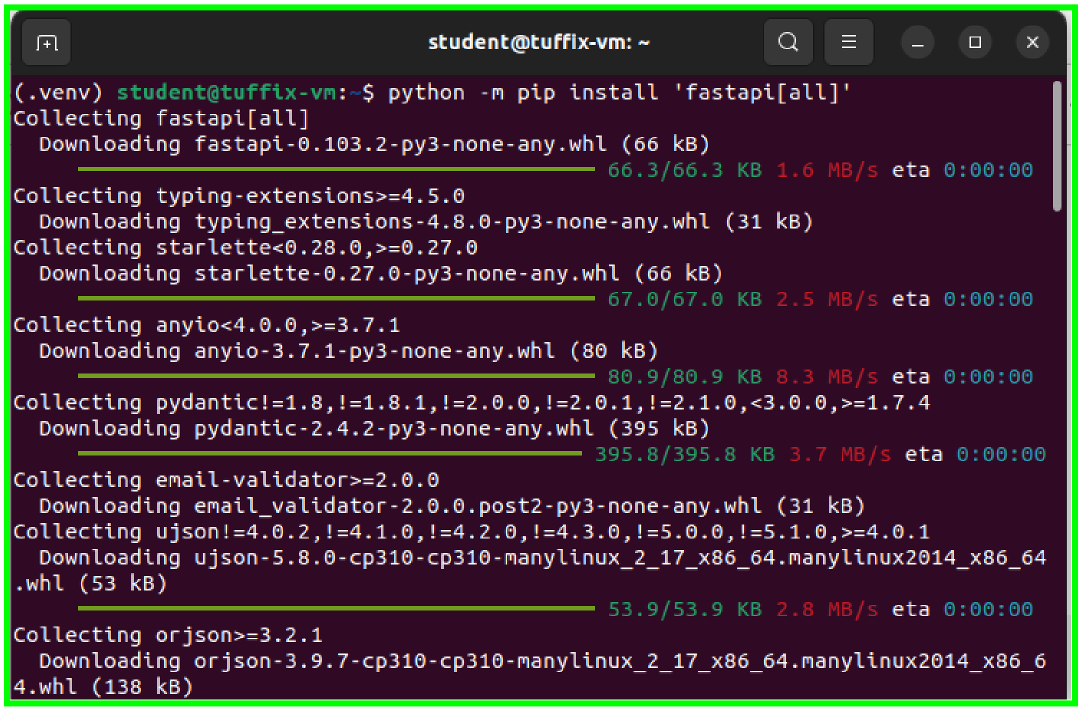
```
(.venv) student@tuffix-vm:~$ sudo apt install --yes ruby-foreman
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gir1.2-snapd-1 python3-debconf update-notifier-common
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  fonts-lato libruby3.0 rake ruby ruby-dotenv ruby-net-telnet ruby-rubygems
  ruby-thor ruby-webrick ruby-xmlrpc ruby3.0 rubygems-integration
Suggested packages:
  ri ruby-dev bundler
The following NEW packages will be installed:
  fonts-lato libruby3.0 rake ruby ruby-dotenv ruby-foreman ruby-net-telnet
  ruby-rubygems ruby-thor ruby-webrick ruby-xmlrpc ruby3.0
  rubygems-integration
0 upgraded, 13 newly installed, 0 to remove and 173 not upgraded.
Need to get 8,330 kB of archives.
After this operation, 37.4 MB of additional disk space will be used.
Get:1 http://gb.archive.ubuntu.com/ubuntu jammy/main amd64 fonts-lato all 2.0-2.
1 [2,696 kB]
Get:2 http://gb.archive.ubuntu.com/ubuntu jammy/main amd64 rubygems-integration
all 1.18 [5,336 B]
Get:3 http://gb.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ruby3.0 amd64
```

Installing HTTPIE



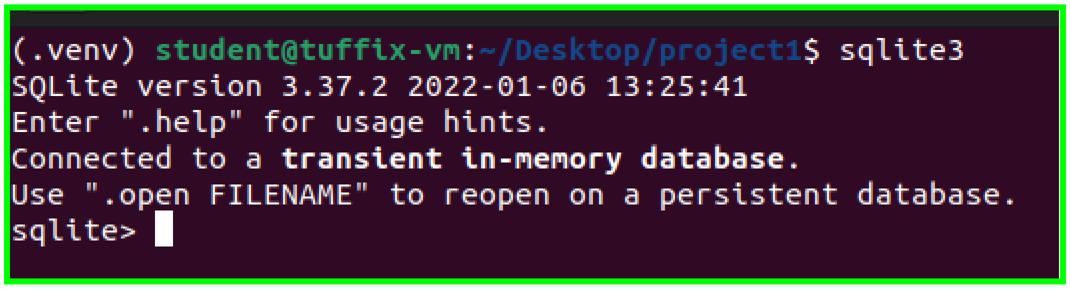
```
(.venv) student@tuffix-vm:~$ sudo snap install httpie
httpie 3.2.2 from Jakub (jakubroztocil) installed
```

Installing FastAPI



```
(.venv) student@tuffix-vm:~$ python -m pip install 'fastapi[all]'
Collecting fastapi[all]
  Downloading fastapi-0.103.2-py3-none-any.whl (66 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 66.3/66.3 KB 1.6 MB/s eta 0:00:00
Collecting typing-extensions>=4.5.0
  Downloading typing_extensions-4.8.0-py3-none-any.whl (31 kB)
Collecting starlette<0.28.0,>=0.27.0
  Downloading starlette-0.27.0-py3-none-any.whl (66 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 67.0/67.0 KB 2.5 MB/s eta 0:00:00
Collecting anyio<4.0.0,>=3.7.1
  Downloading anyio-3.7.1-py3-none-any.whl (80 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 80.9/80.9 KB 8.3 MB/s eta 0:00:00
Collecting pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4
  Downloading pydantic-2.4.2-py3-none-any.whl (395 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 395.8/395.8 KB 3.7 MB/s eta 0:00:00
Collecting email-validator>=2.0.0
  Downloading email_validator-2.0.0.post2-py3-none-any.whl (31 kB)
Collecting ujson!=4.0.2,!=4.1.0,!=4.2.0,!=4.3.0,!=5.0.0,!=5.1.0,>=4.0.1
  Downloading ujson-5.8.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64
.whl (53 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 53.9/53.9 KB 2.8 MB/s eta 0:00:00
Collecting orjson>=3.2.1
  Downloading orjson-3.9.7-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl (138 kB)
```

Installing Sqlite3

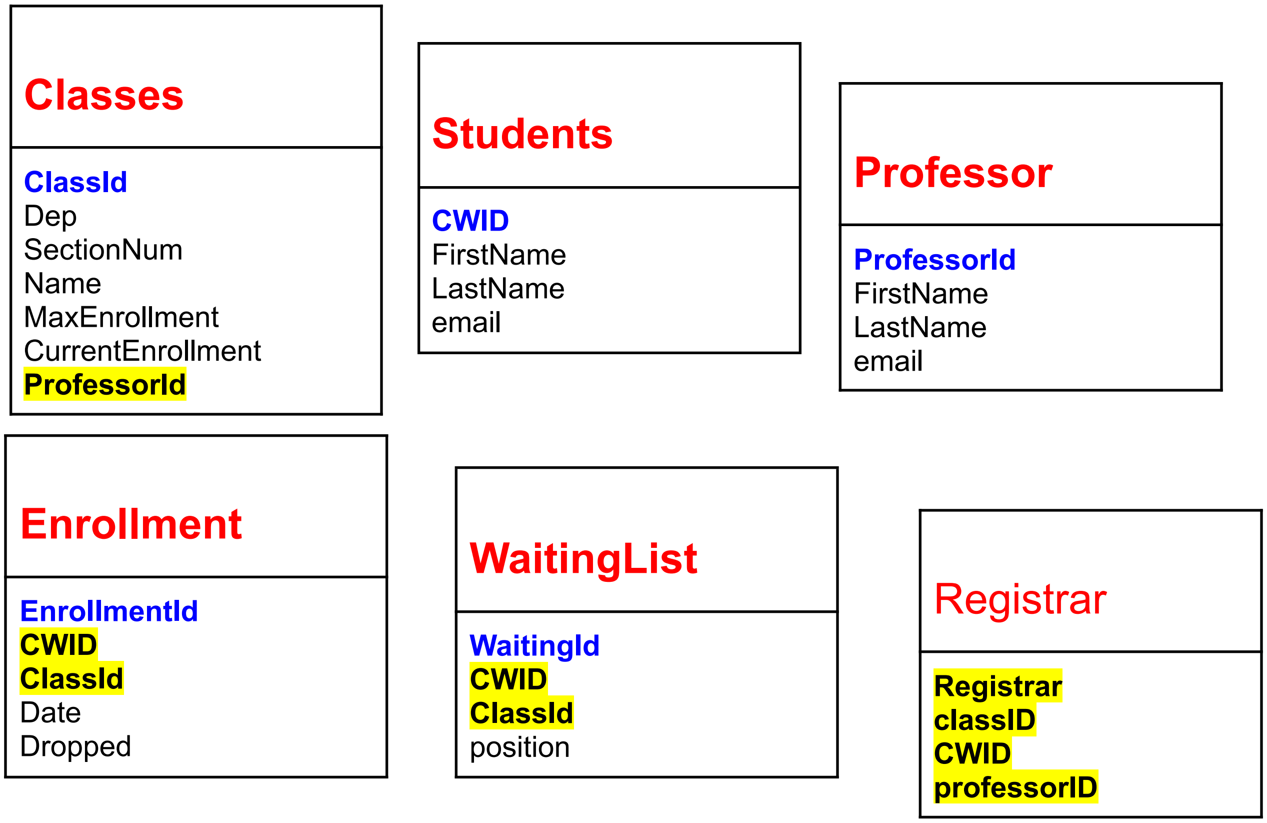


```
(.venv) student@tuffix-vm:~/Desktop/project1$ sqlite3
SQLite version 3.37.2 2022-01-06 13:25:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> 
```

Relationships

KEY

- **Foreign Key**
- **PrimaryKey**



Classes

ClassId
Dep
SectionNum
Name
MaxEnrollment
CurrentEnrollment
ProfessorId

Students

CWID
FirstName
LastName
email

Professor

ProfessorId
FirstName
LastName
email

Enrollment

EnrollmentId
CWID
ClassId
Date
Dropped

WaitingList

WaitingId
CWID
ClassId
position

Registrar

Registrar
classID
CWID
professorID

Define the API

Student

- **GET /students**: **List all the students**

Class

- **GET** /class: **List available classes**
- **POST** /class/enroll: **Attempt to enroll in a class**
- **PUT** /class/drop: **Drop a class**

Enrollment

- **GET** /enrollment: **List of enrollments**

Professor

- **GET** /professor: **List of professors teaching**
- **GET** /professor/{professorID}/class/enrollment: **View current enrollment for their classes**
- **GET** /professor/{professorID}/class/dropped: **View students who have dropped the class**
- **POST** /professor/{professorID}/class/drop_student: **Drop students administratively**

Registrar

- **GET** /registrar/: **List of items that got changed**
- **POST** /registrar/class/add: **Add new classes and sections**
- **DELETE** /registrar/class/remove: **Remove existing sections**
- **POST** /registrar/class/changeProfessor: **Change the instructor for a section**
- **POST** /registrar/waitinglist/{CWID}/enrollment: **Freeze automatic enrollment from waiting lists**

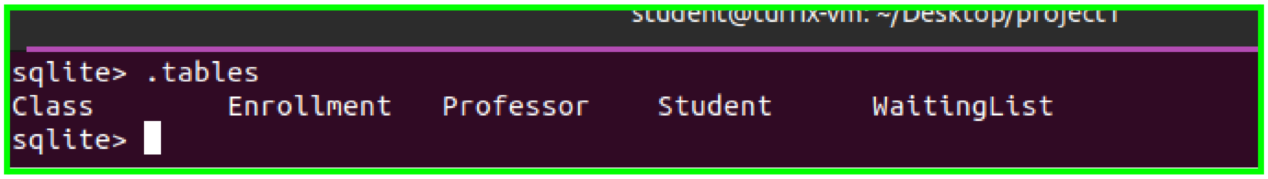
Waiting List

- **GET** /waitnglist: **List of students on the waitlist of each course**
- **GET** /students/{CWID}/waiting_list_position: **View current position on the waiting list**
- **DELETE** /students/{CWID}/waiting_list/remove: **Remove themselves from a waiting list**

- **GET** /professor/{CWID}/classes/waiting_list: **View the current waiting list for the course**

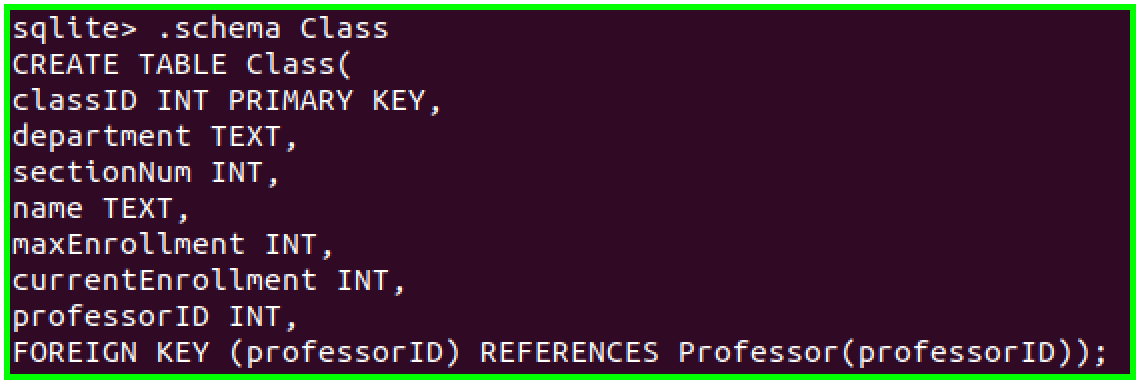
Design the Database

Tables



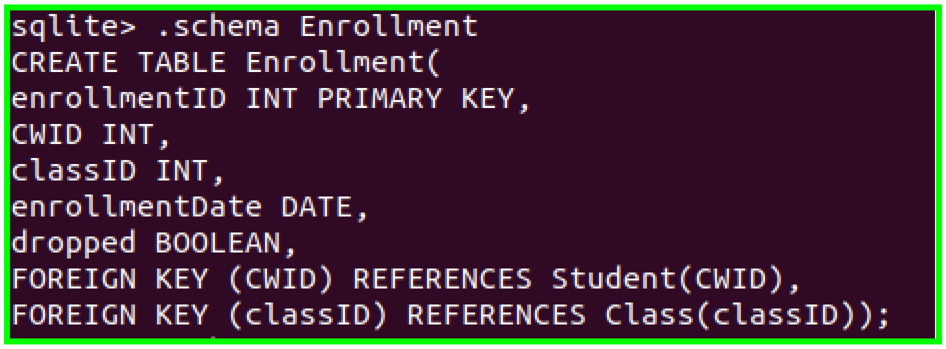
```
sqlite> .tables
Class          Enrollment    Professor     Student       WaitingList
sqlite> 
```

Class Schema



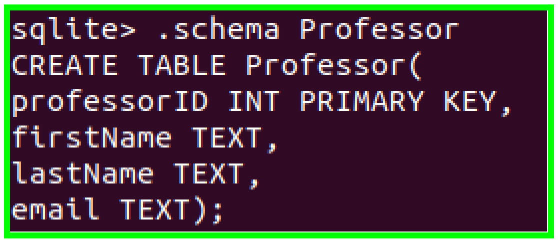
```
sqlite> .schema Class
CREATE TABLE Class(
classID INT PRIMARY KEY,
department TEXT,
sectionNum INT,
name TEXT,
maxEnrollment INT,
currentEnrollment INT,
professorID INT,
FOREIGN KEY (professorID) REFERENCES Professor(professorID));
```

Enrollment Schema



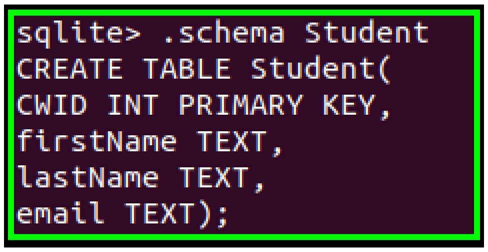
```
sqlite> .schema Enrollment
CREATE TABLE Enrollment(
enrollmentID INT PRIMARY KEY,
CWID INT,
classID INT,
enrollmentDate DATE,
dropped BOOLEAN,
FOREIGN KEY (CWID) REFERENCES Student(CWID),
FOREIGN KEY (classID) REFERENCES Class(classID));
```

Professor Schema



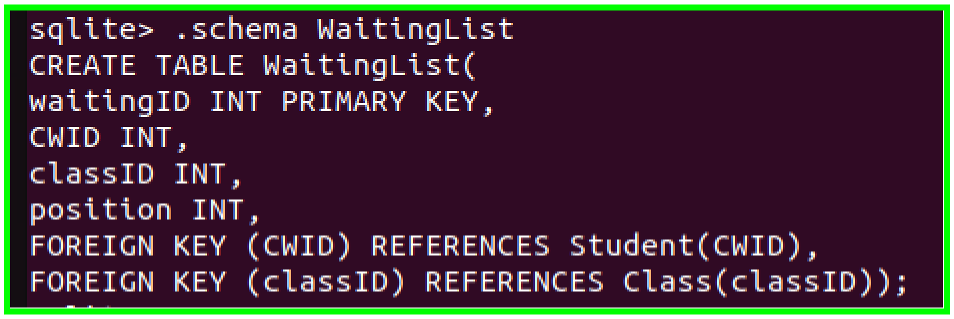
```
sqlite> .schema Professor
CREATE TABLE Professor(
professorID INT PRIMARY KEY,
firstName TEXT,
lastName TEXT,
email TEXT);
```

Student Schema



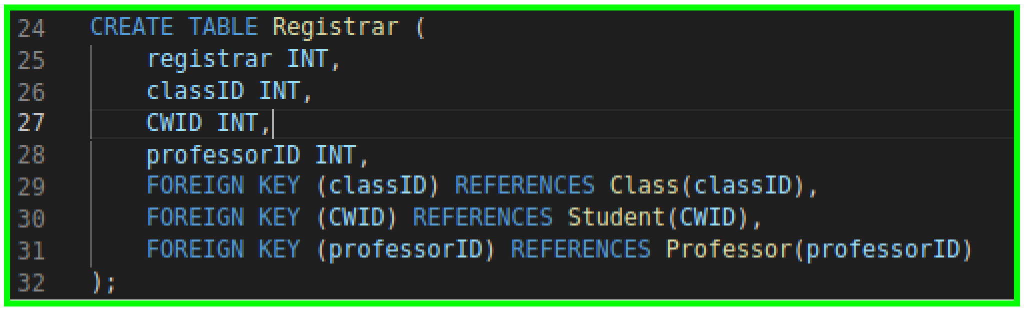
```
sqlite> .schema Student
CREATE TABLE Student(
CWID INT PRIMARY KEY,
firstName TEXT,
lastName TEXT,
email TEXT);
```

WaitingList Schema



```
sqlite> .schema WaitingList
CREATE TABLE WaitingList(
waitingID INT PRIMARY KEY,
CWID INT,
classID INT,
position INT,
FOREIGN KEY (CWID) REFERENCES Student(CWID),
FOREIGN KEY (classID) REFERENCES Class(classID));
```

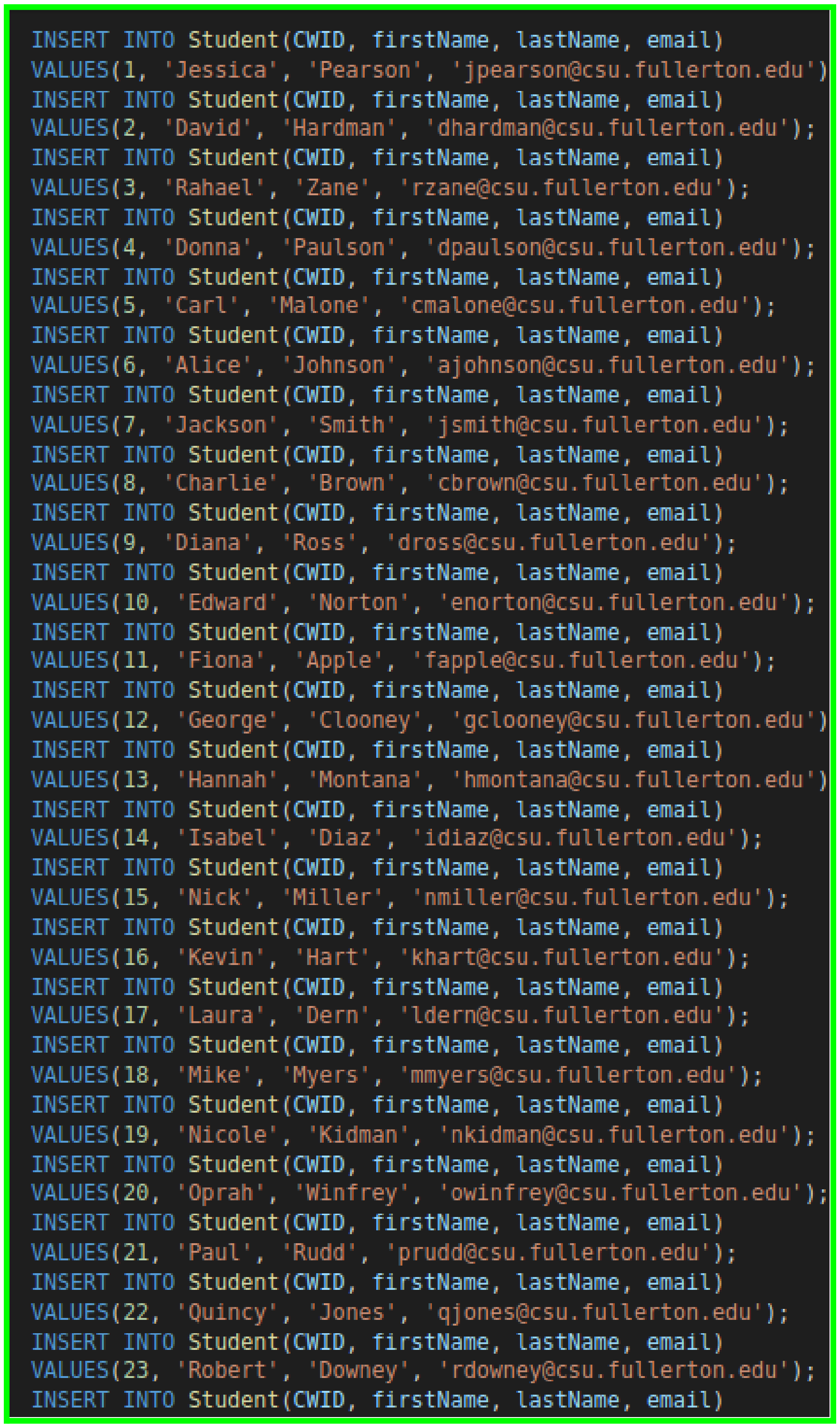
Registrar Schema



```
CREATE TABLE Registrar (
    registrar INT,
    classID INT,
    CWID INT,
    professorID INT,
    FOREIGN KEY (classID) REFERENCES Class(classID),
    FOREIGN KEY (CWID) REFERENCES Student(CWID),
    FOREIGN KEY (professorID) REFERENCES Professor(professorID)
);
```

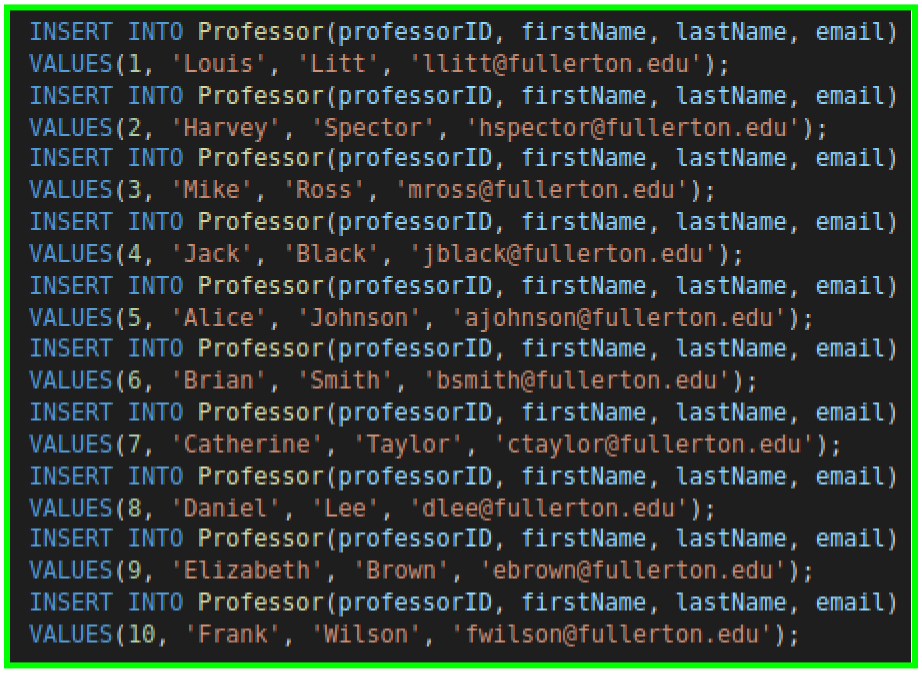
Data Inserted

Student data:



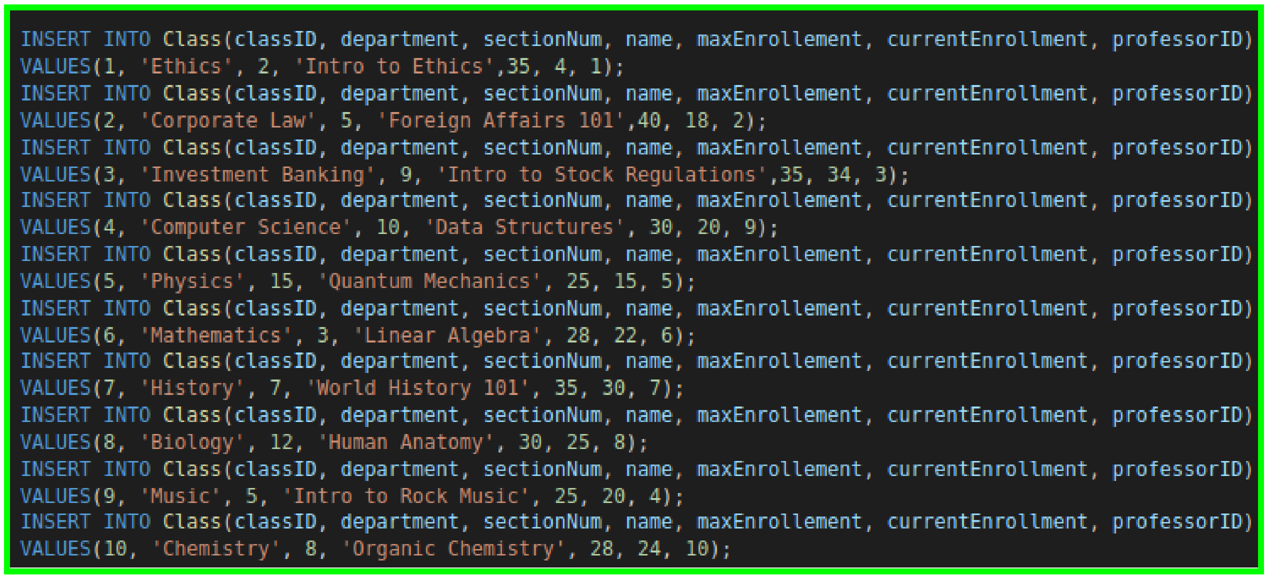
```
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(1, 'Jessica', 'Pearson', 'jpearson@csu.fullerton.edu')
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(2, 'David', 'Hardman', 'dhardman@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(3, 'Rahael', 'Zane', 'rzane@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(4, 'Donna', 'Paulson', 'dpaulson@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(5, 'Carl', 'Malone', 'cmalone@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(6, 'Alice', 'Johnson', 'ajohnson@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(7, 'Jackson', 'Smith', 'jsmith@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(8, 'Charlie', 'Brown', 'cbrown@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(9, 'Diana', 'Ross', 'dross@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(10, 'Edward', 'Norton', 'enorton@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(11, 'Fiona', 'Apple', 'fapple@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(12, 'George', 'Clooney', 'gclooney@csu.fullerton.edu')
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(13, 'Hannah', 'Montana', 'hmontana@csu.fullerton.edu')
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(14, 'Isabel', 'Diaz', 'idiaz@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(15, 'Nick', 'Miller', 'nmiller@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(16, 'Kevin', 'Hart', 'khart@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(17, 'Laura', 'Dern', 'ldern@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(18, 'Mike', 'Myers', 'mmyers@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(19, 'Nicole', 'Kidman', 'nkidman@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(20, 'Oprah', 'Winfrey', 'owinfrey@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(21, 'Paul', 'Rudd', 'prudd@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(22, 'Quincy', 'Jones', 'qjones@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
VALUES(23, 'Robert', 'Downey', 'rdowney@csu.fullerton.edu');
INSERT INTO Student(CWID, firstName, lastName, email)
```

Professor data:



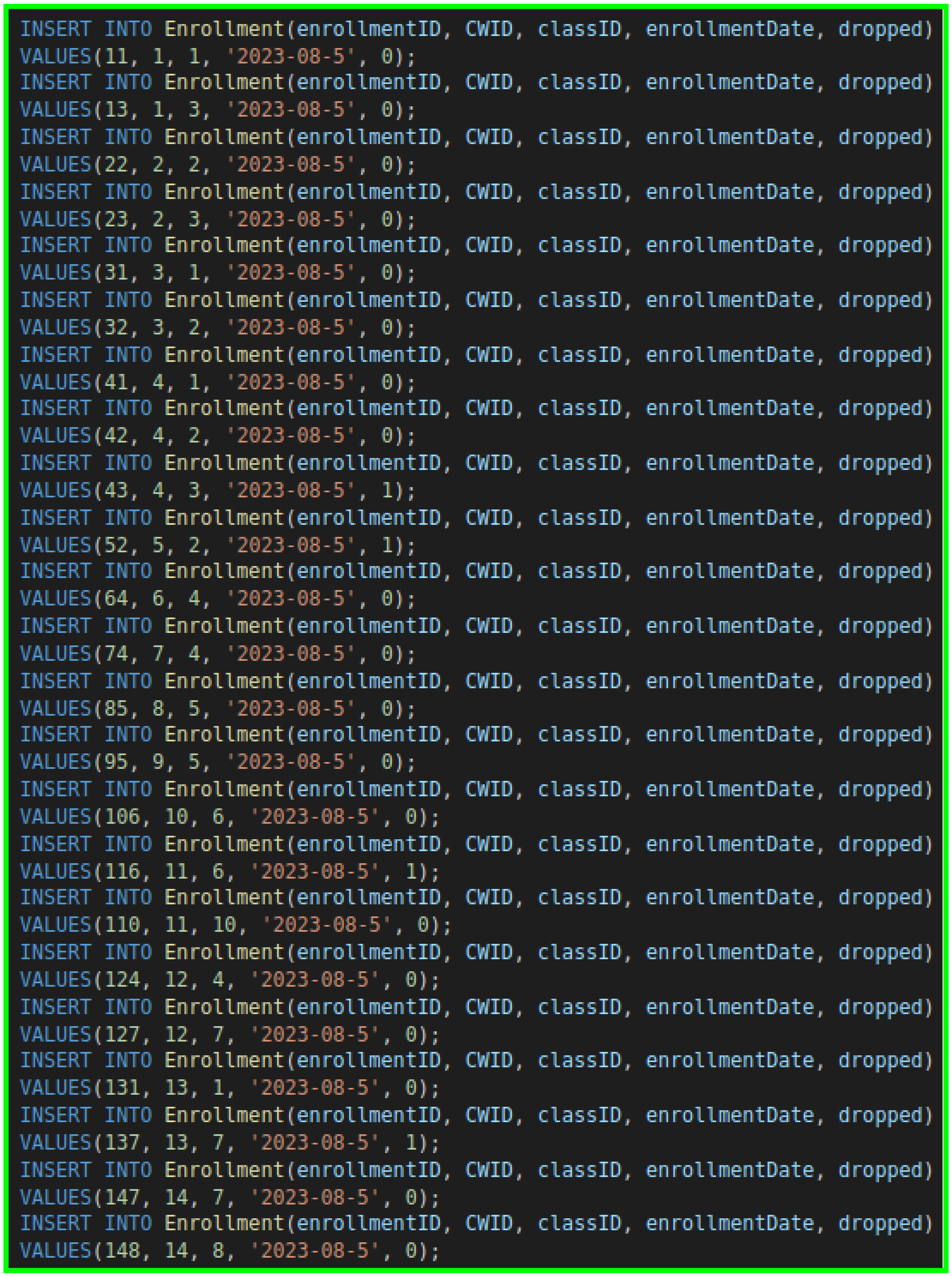
```sql
INSERT INTO Professor(professorID, firstName, lastName, email)
VALUES(1, 'Louis', 'Litt', 'llitt@fullerton.edu');
INSERT INTO Professor(professorID, firstName, lastName, email)
VALUES(2, 'Harvey', 'Spector', 'hspector@fullerton.edu');
INSERT INTO Professor(professorID, firstName, lastName, email)
VALUES(3, 'Mike', 'Ross', 'mross@fullerton.edu');
INSERT INTO Professor(professorID, firstName, lastName, email)
VALUES(4, 'Jack', 'Black', 'jblack@fullerton.edu');
INSERT INTO Professor(professorID, firstName, lastName, email)
VALUES(5, 'Alice', 'Johnson', 'ajohnson@fullerton.edu');
INSERT INTO Professor(professorID, firstName, lastName, email)
VALUES(6, 'Brian', 'Smith', 'bsmith@fullerton.edu');
INSERT INTO Professor(professorID, firstName, lastName, email)
VALUES(7, 'Catherine', 'Taylor', 'ctaylor@fullerton.edu');
INSERT INTO Professor(professorID, firstName, lastName, email)
VALUES(8, 'Daniel', 'Lee', 'dlee@fullerton.edu');
INSERT INTO Professor(professorID, firstName, lastName, email)
VALUES(9, 'Elizabeth', 'Brown', 'ebrown@fullerton.edu');
INSERT INTO Professor(professorID, firstName, lastName, email)
VALUES(10, 'Frank', 'Wilson', 'fwilson@fullerton.edu');
```

Class Data:



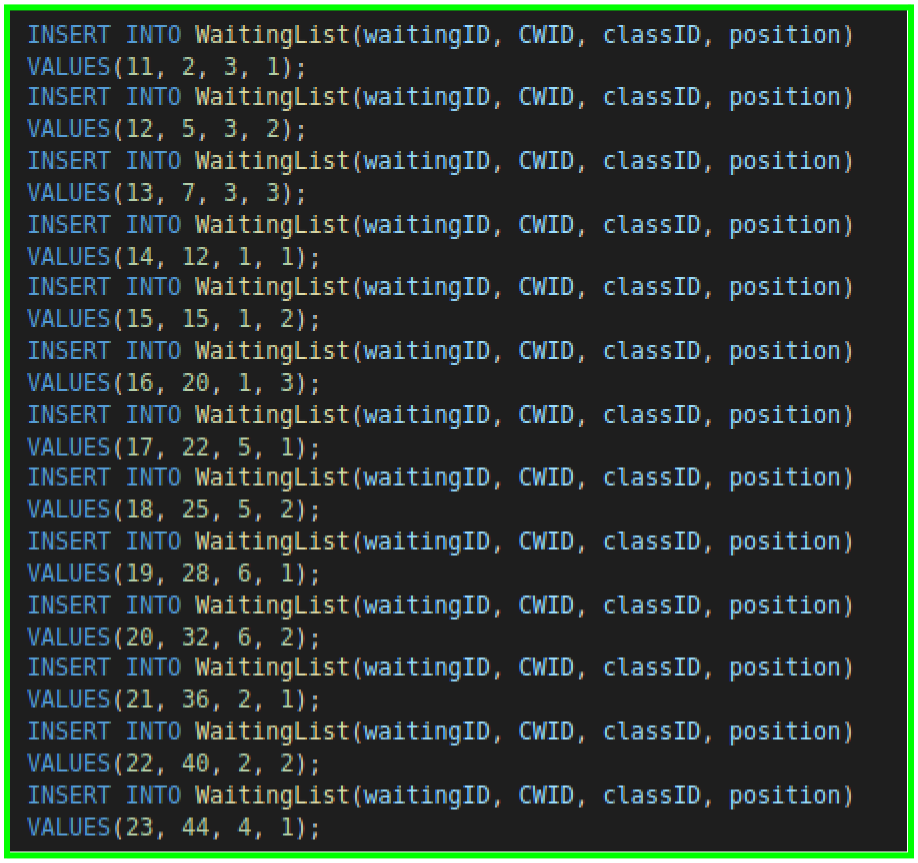
```sql
INSERT INTO Class(classID, department, sectionNum, name, maxEnrollement, currentEnrollment, professorID)
VALUES(1, 'Ethics', 2, 'Intro to Ethics',35, 4, 1);
INSERT INTO Class(classID, department, sectionNum, name, maxEnrollement, currentEnrollment, professorID)
VALUES(2, 'Corporate Law', 5, 'Foreign Affairs 101',40, 18, 2);
INSERT INTO Class(classID, department, sectionNum, name, maxEnrollement, currentEnrollment, professorID)
VALUES(3, 'Investment Banking', 9, 'Intro to Stock Regulations',35, 34, 3);
INSERT INTO Class(classID, department, sectionNum, name, maxEnrollement, currentEnrollment, professorID)
VALUES(4, 'Computer Science', 10, 'Data Structures', 30, 20, 9);
INSERT INTO Class(classID, department, sectionNum, name, maxEnrollement, currentEnrollment, professorID)
VALUES(5, 'Physics', 15, 'Quantum Mechanics', 25, 15, 5);
INSERT INTO Class(classID, department, sectionNum, name, maxEnrollement, currentEnrollment, professorID)
VALUES(6, 'Mathematics', 3, 'Linear Algebra', 28, 22, 6);
INSERT INTO Class(classID, department, sectionNum, name, maxEnrollement, currentEnrollment, professorID)
VALUES(7, 'History', 7, 'World History 101', 35, 30, 7);
INSERT INTO Class(classID, department, sectionNum, name, maxEnrollement, currentEnrollment, professorID)
VALUES(8, 'Biology', 12, 'Human Anatomy', 30, 25, 8);
INSERT INTO Class(classID, department, sectionNum, name, maxEnrollement, currentEnrollment, professorID)
VALUES(9, 'Music', 5, 'Intro to Rock Music', 25, 20, 4);
INSERT INTO Class(classID, department, sectionNum, name, maxEnrollement, currentEnrollment, professorID)
VALUES(10, 'Chemistry', 8, 'Organic Chemistry', 28, 24, 10);
```

Enrollment Data:



```sql
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(11, 1, 1, '2023-08-5', 0);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(13, 1, 3, '2023-08-5', 0);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(22, 2, 2, '2023-08-5', 0);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(23, 2, 3, '2023-08-5', 0);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(31, 3, 1, '2023-08-5', 0);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(32, 3, 2, '2023-08-5', 0);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(41, 4, 1, '2023-08-5', 0);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(42, 4, 2, '2023-08-5', 0);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(43, 4, 3, '2023-08-5', 1);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(52, 5, 2, '2023-08-5', 1);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(64, 6, 4, '2023-08-5', 0);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(74, 7, 4, '2023-08-5', 0);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(85, 8, 5, '2023-08-5', 0);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(95, 9, 5, '2023-08-5', 0);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(106, 10, 6, '2023-08-5', 0);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(116, 11, 6, '2023-08-5', 1);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(110, 11, 10, '2023-08-5', 0);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(124, 12, 4, '2023-08-5', 0);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(127, 12, 7, '2023-08-5', 0);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(131, 13, 1, '2023-08-5', 0);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(137, 13, 7, '2023-08-5', 1);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(147, 14, 7, '2023-08-5', 0);
INSERT INTO Enrollment(enrollmentID, CWID, classID, enrollmentDate, dropped)
VALUES(148, 14, 8, '2023-08-5', 0);
```

Wait List Data:



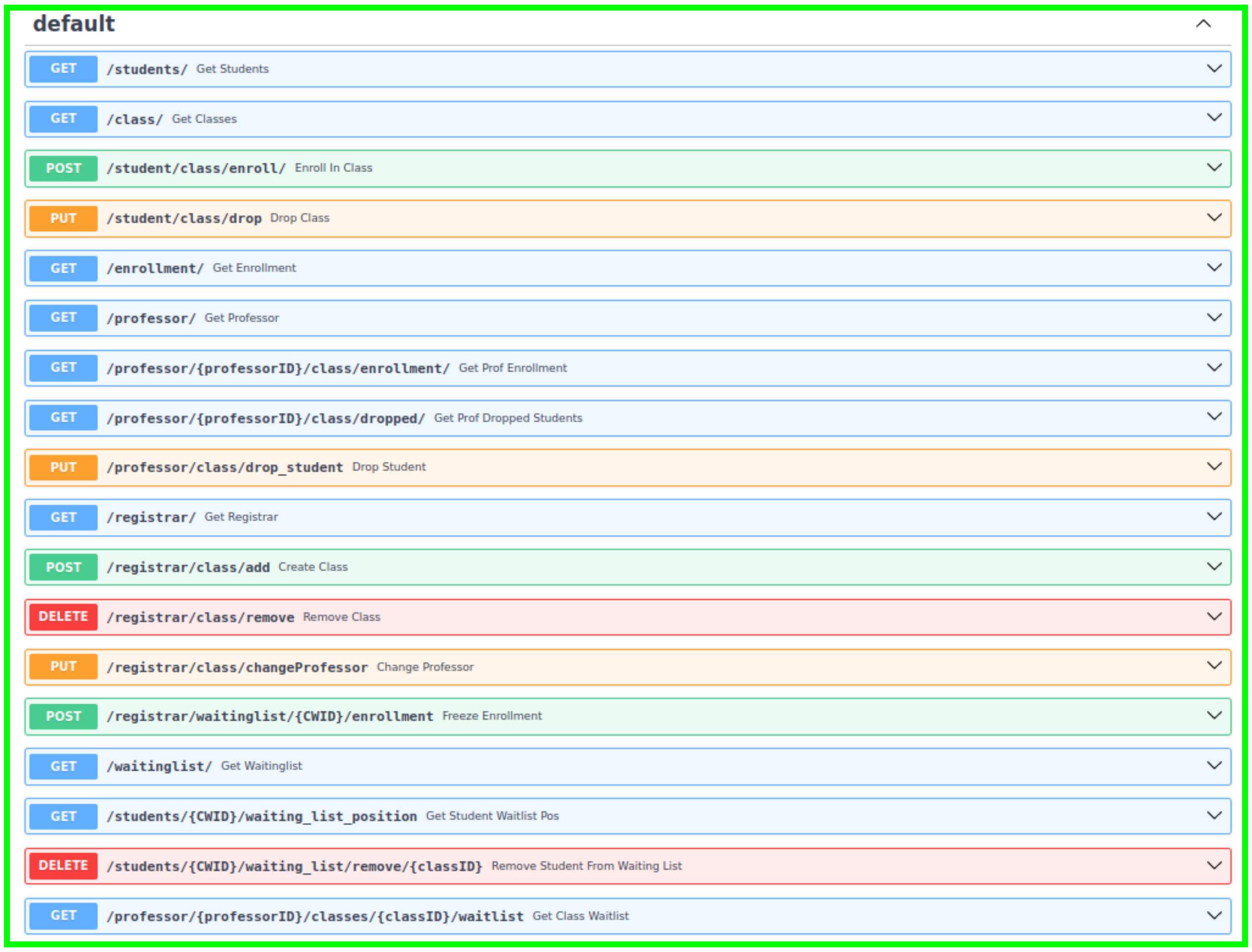
```sql
INSERT INTO WaitingList(waitingID, CWID, classID, position)
VALUES(11, 2, 3, 1);
INSERT INTO WaitingList(waitingID, CWID, classID, position)
VALUES(12, 5, 3, 2);
INSERT INTO WaitingList(waitingID, CWID, classID, position)
VALUES(13, 7, 3, 3);
INSERT INTO WaitingList(waitingID, CWID, classID, position)
VALUES(14, 12, 1, 1);
INSERT INTO WaitingList(waitingID, CWID, classID, position)
VALUES(15, 15, 1, 2);
INSERT INTO WaitingList(waitingID, CWID, classID, position)
VALUES(16, 20, 1, 3);
INSERT INTO WaitingList(waitingID, CWID, classID, position)
VALUES(17, 22, 5, 1);
INSERT INTO WaitingList(waitingID, CWID, classID, position)
VALUES(18, 25, 5, 2);
INSERT INTO WaitingList(waitingID, CWID, classID, position)
VALUES(19, 28, 6, 1);
INSERT INTO WaitingList(waitingID, CWID, classID, position)
VALUES(20, 32, 6, 2);
INSERT INTO WaitingList(waitingID, CWID, classID, position)
VALUES(21, 36, 2, 1);
INSERT INTO WaitingList(waitingID, CWID, classID, position)
VALUES(22, 40, 2, 2);
INSERT INTO WaitingList(waitingID, CWID, classID, position)
VALUES(23, 44, 4, 1);
```

Context: This waitlist data was only used in the beginning for testing purposes. The wait list table for the program inserts data dynamically (initalized empty) whenever a student is added to the waitlist automatically.

FASTAPI

DOCS

- http://127.0.0.1:5000/docs



default

GET /students/ Get Students

GET /class/ Get Classes

POST /student/class/enroll/ Enroll In Class

PUT /student/class/drop Drop Class

GET /enrollment/ Get Enrollment

GET /professor/ Get Professor

GET /professor/{professorID}/class/enrollment/ Get Prof Enrollment

GET /professor/{professorID}/class/dropped/ Get Prof Dropped Students

PUT /professor/class/drop_student Drop Student

GET /registrar/ Get Registrar

POST /registrar/class/add Create Class

DELETE /registrar/class/remove Remove Class

PUT /registrar/class/changeProfessor Change Professor

POST /registrar/waitinglist/{CWID}/enrollment Freeze Enrollment

GET /waitinglist/ Get Waitinglist

GET /students/{CWID}/waiting_list_position Get Student Waitlist Pos

DELETE /students/{CWID}/waiting_list/remove/{classID} Remove Student From Waiting List

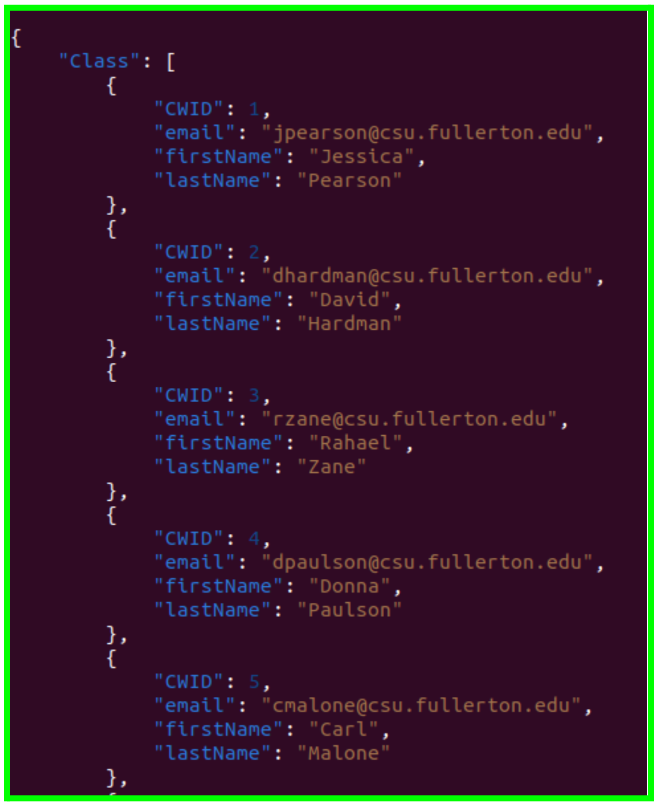
GET /professor/{professorID}/classes/{classID}/waitlist Get Class Waitlist

Context: FastAPI docs shows all of our endpoints that we created

Students

Result GET Student

- http GET http://127.0.0.1:5000/students/

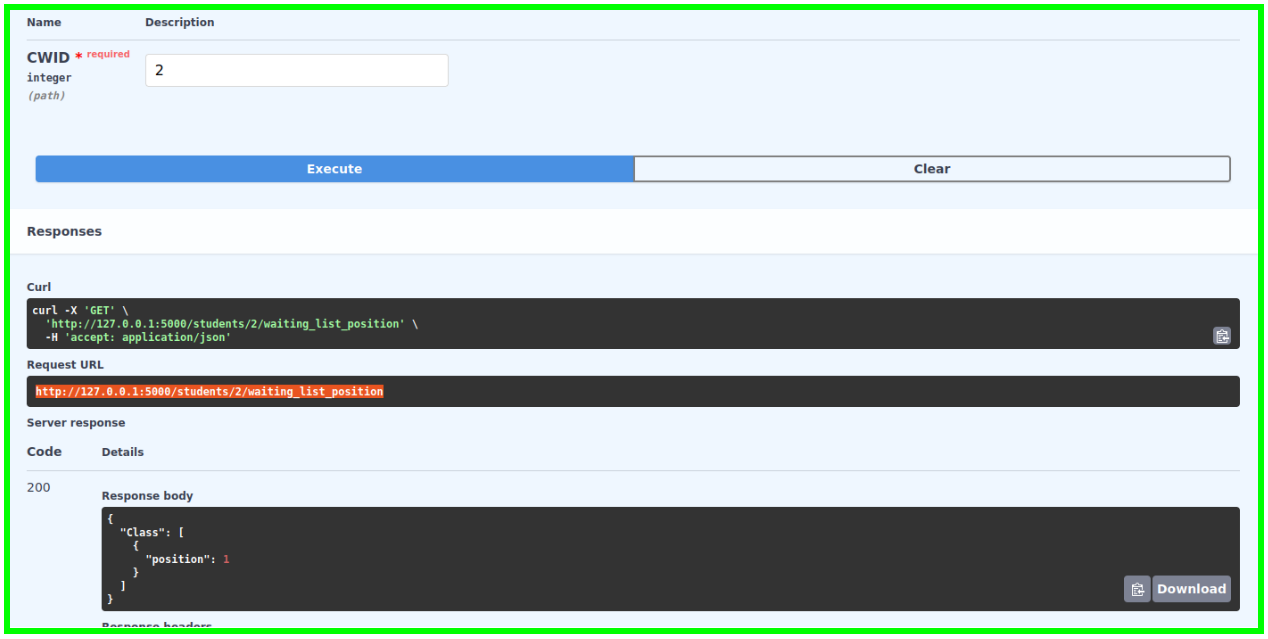


```
{
    "Class": [
        {
            "CWID": 1,
            "email": "jpearson@csu.fullerton.edu",
            "firstName": "Jessica",
            "lastName": "Pearson"
        },
        {
            "CWID": 2,
            "email": "dhardman@csu.fullerton.edu",
            "firstName": "David",
            "lastName": "Hardman"
        },
        {
            "CWID": 3,
            "email": "rzane@csu.fullerton.edu",
            "firstName": "Rahael",
            "lastName": "Zane"
        },
        {
            "CWID": 4,
            "email": "dpaulson@csu.fullerton.edu",
            "firstName": "Donna",
            "lastName": "Paulson"
        },
        {
            "CWID": 5,
            "email": "cmalone@csu.fullerton.edu",
            "firstName": "Carl",
            "lastName": "Malone"
        },
```

Context: Our database has a total of 50 students

Result GET Student Waitlist Position

- http GET http://127.0.0.1:5000/students/2/waiting_list_position


Name Description

CWID * required
integer
(path)
2

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:5000/students/2/waiting_list_position' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:5000/students/2/waiting_list_position
```

Server response

Code Details

200

Response body

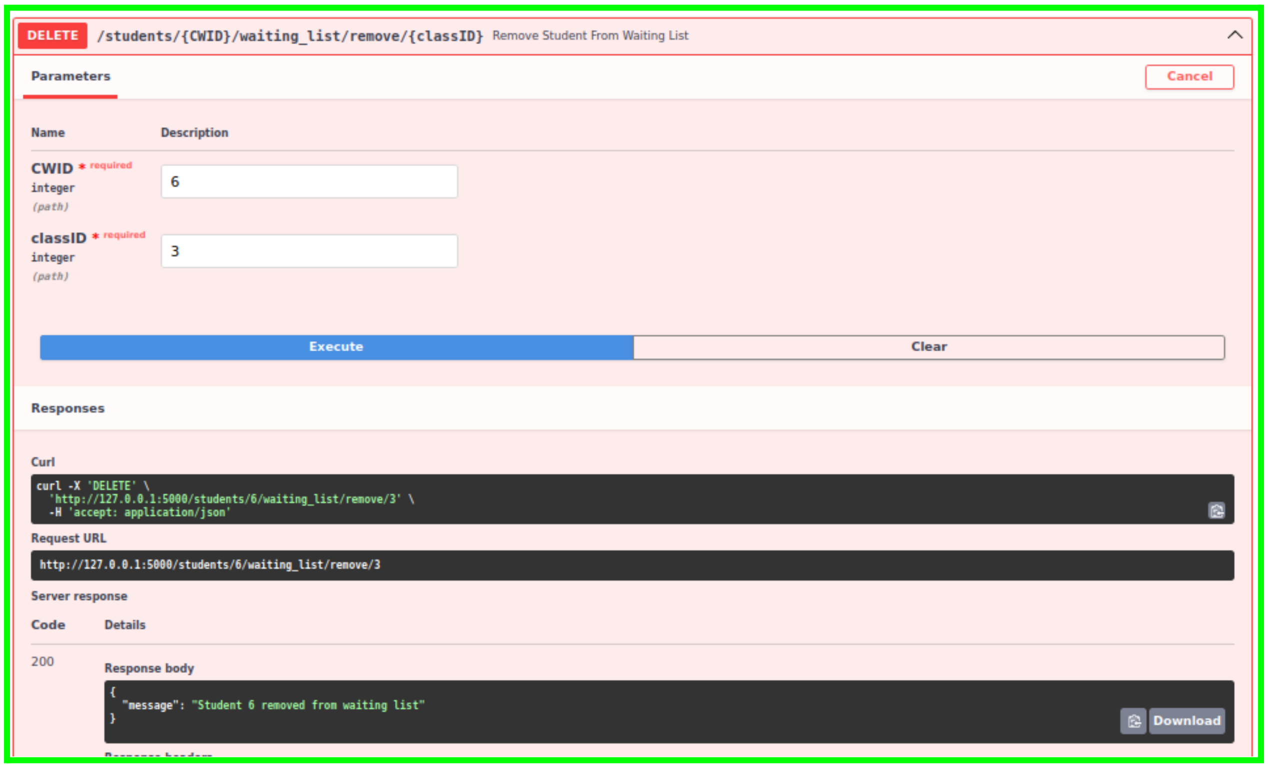
```
{
  "Class": [
    {
      "position": 1
    }
  ]
}
```

Download

Context: Students can view their waitlist position for a class

Result DELETE Student Waitlist Position

- http GET http://127.0.0.1:5000/students/6/waiting_list/remove/3


DELETE /students/{CWID}/waiting_list/remove/{classID} Remove Student From Waiting List

Parameters Cancel

Name Description

CWID * required
integer
(path)
6

classID * required
integer
(path)
3

Execute Clear

Responses

Curl

```
curl -X 'DELETE' \
  'http://127.0.0.1:5000/students/6/waiting_list/remove/3' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:5000/students/6/waiting_list/remove/3
```

Server response

Code Details

200

Response body

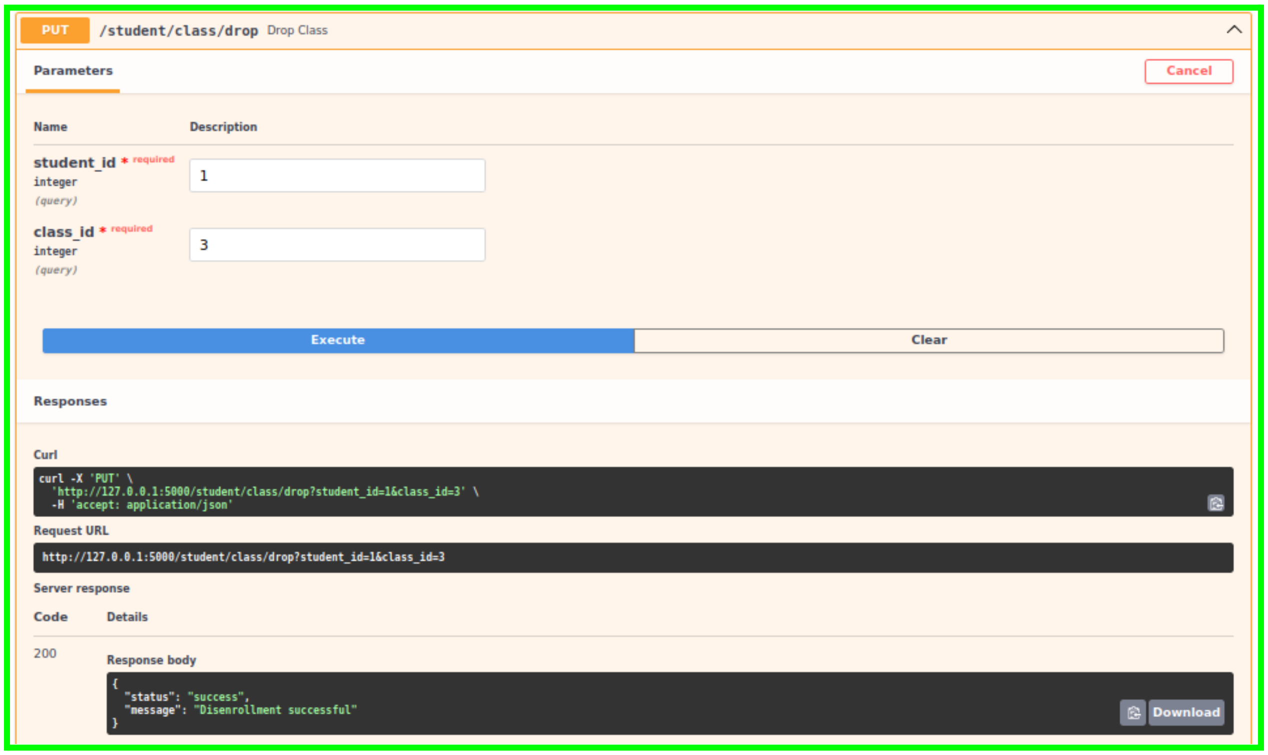
```
{
  "message": "Student 6 removed from waiting list"
}
```

Download

Context: Students can remove their waitlist position for a class.

Result PUT Student Class Drop

- http://127.0.0.1:5000/student/class/drop?student_id=1&class_id=3



student_id: 1

class_id: 3

Curl

```
curl -X 'PUT' \
  'http://127.0.0.1:5000/student/class/drop?student_id=1&class_id=3' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:5000/student/class/drop?student_id=1&class_id=3
```

Server response

Code Details

200 Response body

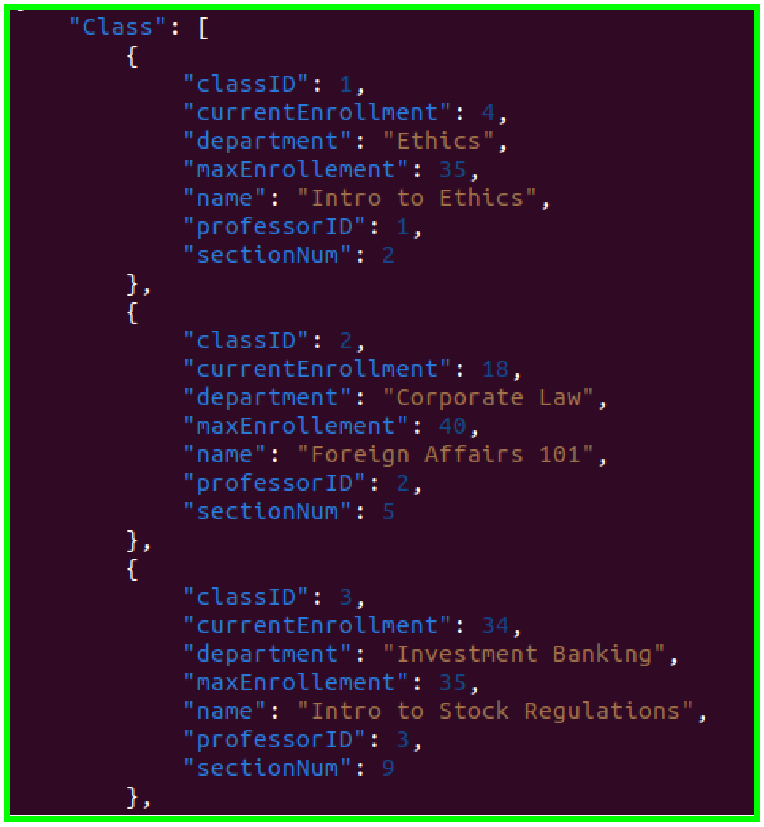
```
{
  "status": "success",
  "message": "Disenrollment successful"
}
```

Context: Students can remove a class from their schedule.

Class

Result GET Class

- http GET http://127.0.0.1:5000/class/

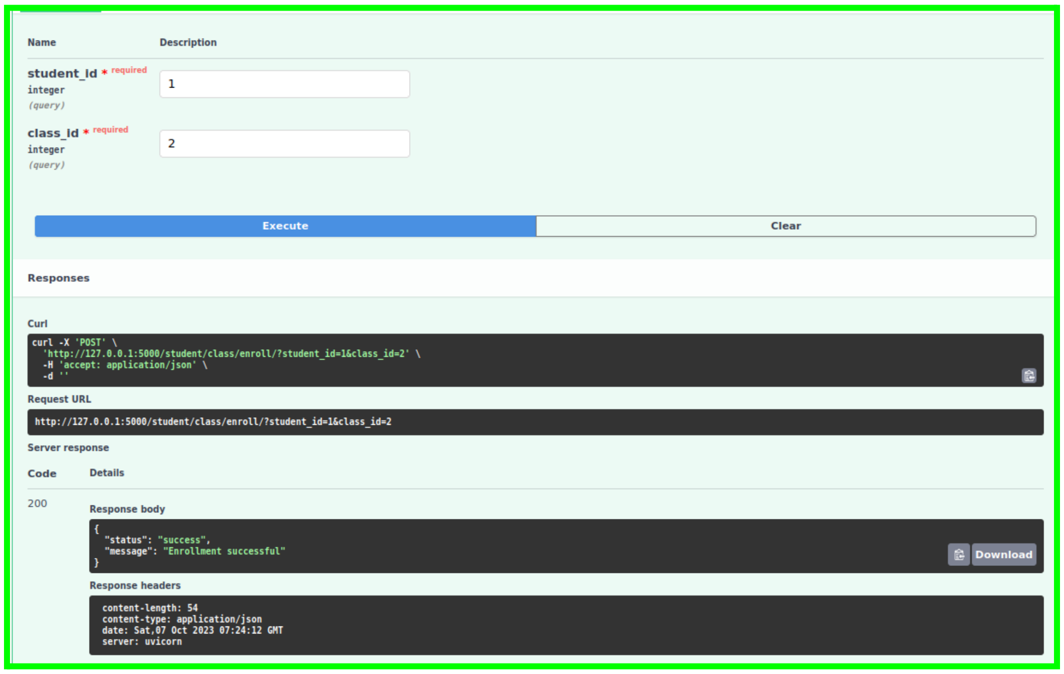


```
    "Class": [
        {
            "classID": 1,
            "currentEnrollment": 4,
            "department": "Ethics",
            "maxEnrollement": 35,
            "name": "Intro to Ethics",
            "professorID": 1,
            "sectionNum": 2
        },
        {
            "classID": 2,
            "currentEnrollment": 18,
            "department": "Corporate Law",
            "maxEnrollement": 40,
            "name": "Foreign Affairs 101",
            "professorID": 2,
            "sectionNum": 5
        },
        {
            "classID": 3,
            "currentEnrollment": 34,
            "department": "Investment Banking",
            "maxEnrollement": 35,
            "name": "Intro to Stock Regulations",
            "professorID": 3,
            "sectionNum": 9
        },
```

Context: Our database has a total of 10 different classes

Result POST Enroll & PUT Drop

- http://127.0.0.1:5000/student/class/enroll/?student_id=1&class_id=2



Name | Description
student_id * required integer (query) | 1
class_id * required integer (query) | 2

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:5000/student/class/enroll/?student_id=1&class_id=2' \
  -H 'accept: application/json' \
  -d ''
```

Request URL

```
http://127.0.0.1:5000/student/class/enroll/?student_id=1&class_id=2
```

Server response

Code: 200

Response body

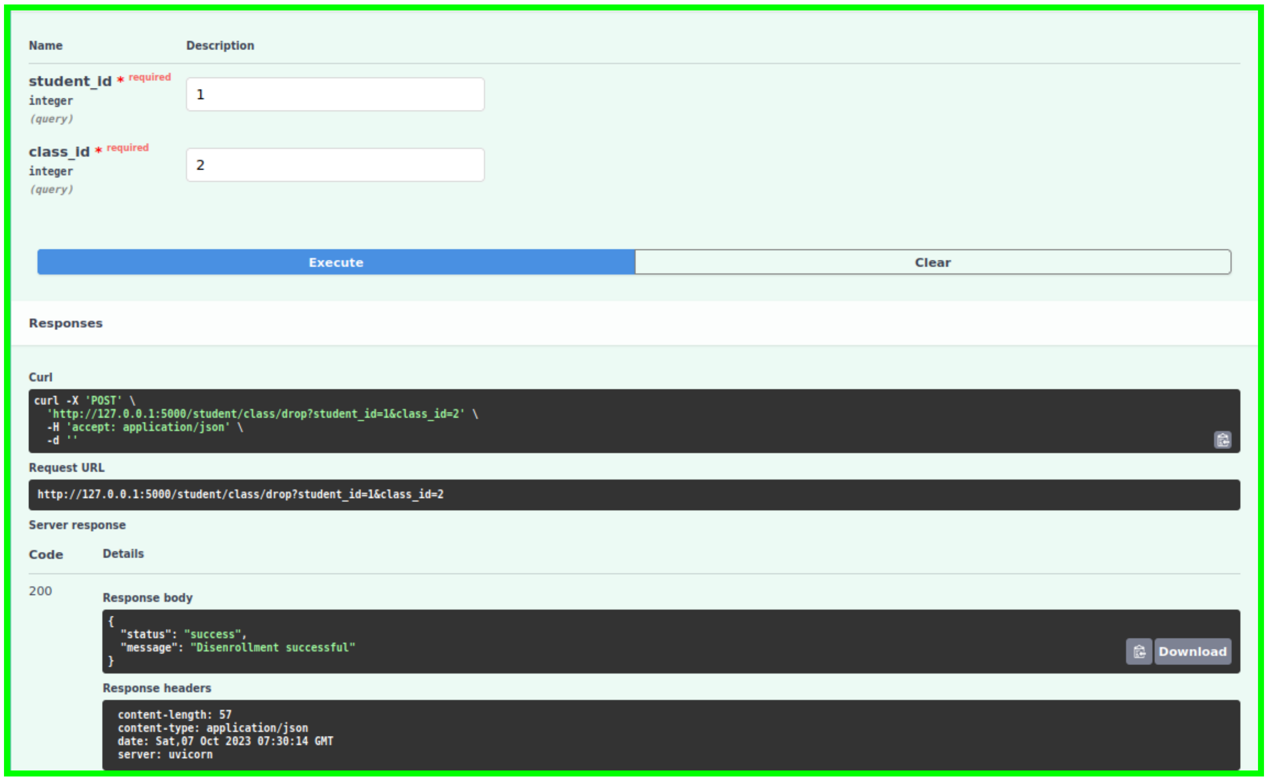
```
{
  "status": "success",
  "message": "Enrollment successful"
}
```

Response headers

```
content-length: 54
content-type: application/json
date: Sat,07 Oct 2023 07:24:12 GMT
server: uvicorn
```

Context: Enrolled the student into Corporate Law

- http://127.0.0.1:5000/student/class/drop?student_id=1&class_id=2



Name | Description
student_id * required integer (query) | 1
class_id * required integer (query) | 2

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:5000/student/class/drop?student_id=1&class_id=2' \
  -H 'accept: application/json' \
  -d ''
```

Request URL

```
http://127.0.0.1:5000/student/class/drop?student_id=1&class_id=2
```

Server response

Code: 200

Response body

```
{
  "status": "success",
  "message": "Disenrollment successful"
}
```

Response headers

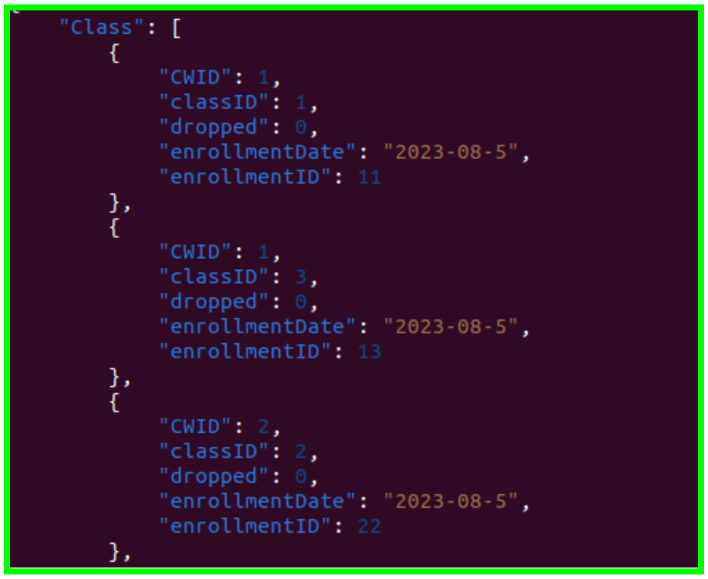
```
content-length: 57
content-type: application/json
date: Sat,07 Oct 2023 07:30:14 GMT
server: uvicorn
```

Context: Dropped the student from Corporate Law

Enrollment

Result GET Enrollment

- http GET http://127.0.0.1:5000/enrollment/



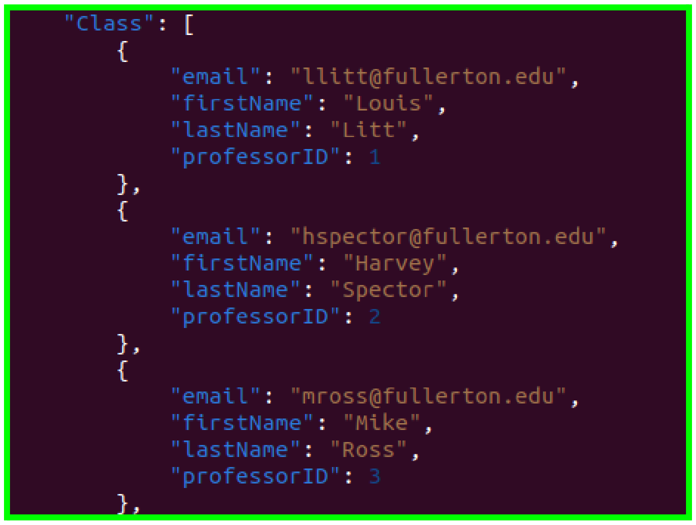
```
    "Class": [
        {
            "CWID": 1,
            "classID": 1,
            "dropped": 0,
            "enrollmentDate": "2023-08-5",
            "enrollmentID": 11
        },
        {
            "CWID": 1,
            "classID": 3,
            "dropped": 0,
            "enrollmentDate": "2023-08-5",
            "enrollmentID": 13
        },
        {
            "CWID": 2,
            "classID": 2,
            "dropped": 0,
            "enrollmentDate": "2023-08-5",
            "enrollmentID": 22
        },
```

Context: Our database has enrolled 50 students into different classes

Professor

Result GET Professor, Pro Enrollment, Prof Dropped Students

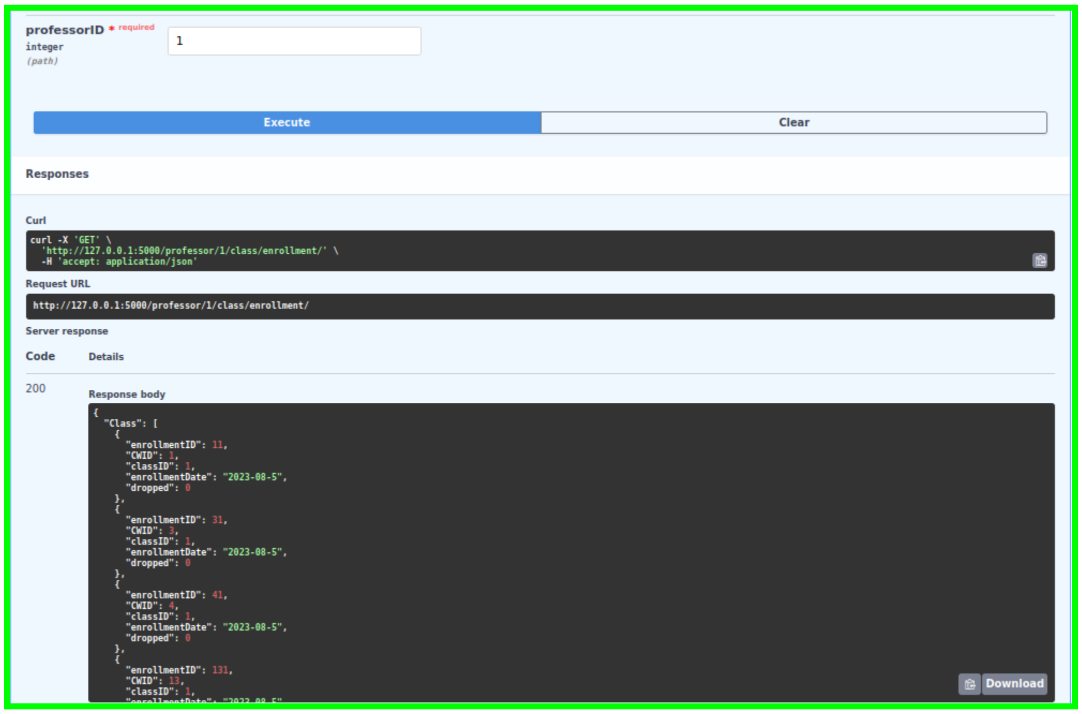
- http GET http://127.0.0.1:5000/professor/



```
    "Class": [
        {
            "email": "llitt@fullerton.edu",
            "firstName": "Louis",
            "lastName": "Litt",
            "professorID": 1
        },
        {
            "email": "hspector@fullerton.edu",
            "firstName": "Harvey",
            "lastName": "Spector",
            "professorID": 2
        },
        {
            "email": "mross@fullerton.edu",
            "firstName": "Mike",
            "lastName": "Ross",
            "professorID": 3
        },
```

Contex: Our database has a total of 10 different Professors

- http://127.0.0.1:5000/professor/1/class/enrollment/



Curl

```
curl -X 'GET' \
  'http://127.0.0.1:5000/professor/1/class/enrollment/' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:5000/professor/1/class/enrollment/
```

Server response

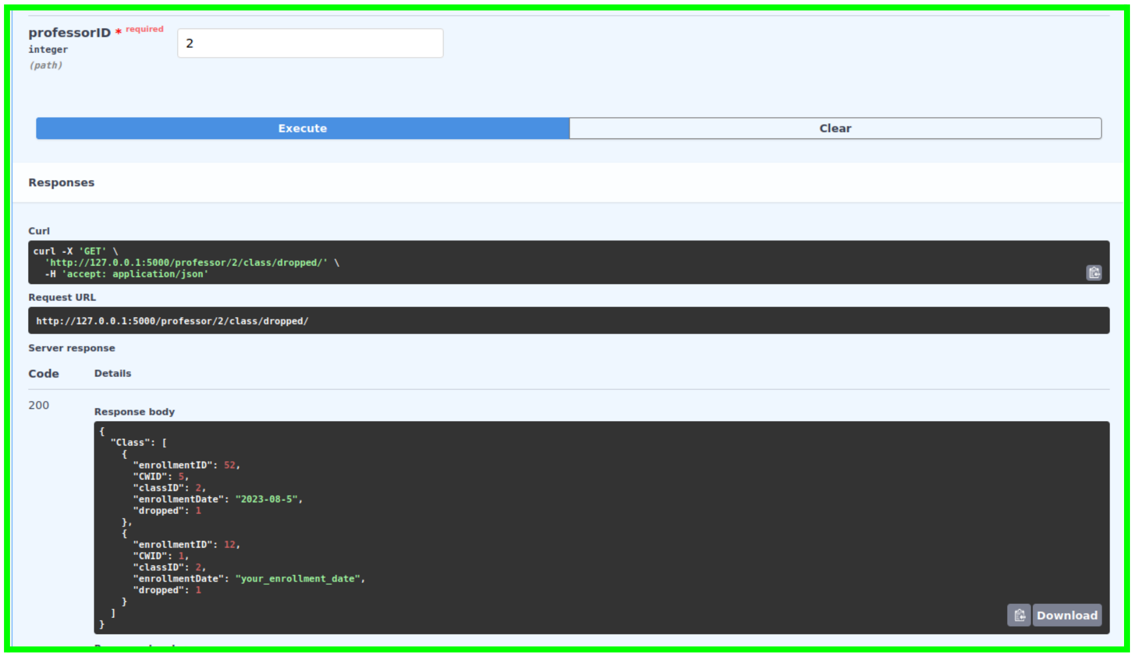
Code Details

200 Response body

```
{
  "Class": [
    {
      "enrollmentID": 11,
      "CWID": 1,
      "classID": 1,
      "enrollmentDate": "2023-08-5",
      "dropped": 0
    },
    {
      "enrollmentID": 31,
      "CWID": 3,
      "classID": 1,
      "enrollmentDate": "2023-08-5",
      "dropped": 0
    },
    {
      "enrollmentID": 41,
      "CWID": 4,
      "classID": 1,
      "enrollmentDate": "2023-08-5",
      "dropped": 0
    },
    {
      "enrollmentID": 131,
      "CWID": 13,
      "classID": 1,
```

Contex: View the current enrollment for their classes based on the Professor

- http://127.0.0.1:5000/professor/2/class/dropped/



Curl

```
curl -X 'GET' \
  'http://127.0.0.1:5000/professor/2/class/dropped/' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:5000/professor/2/class/dropped/
```

Server response

Code Details

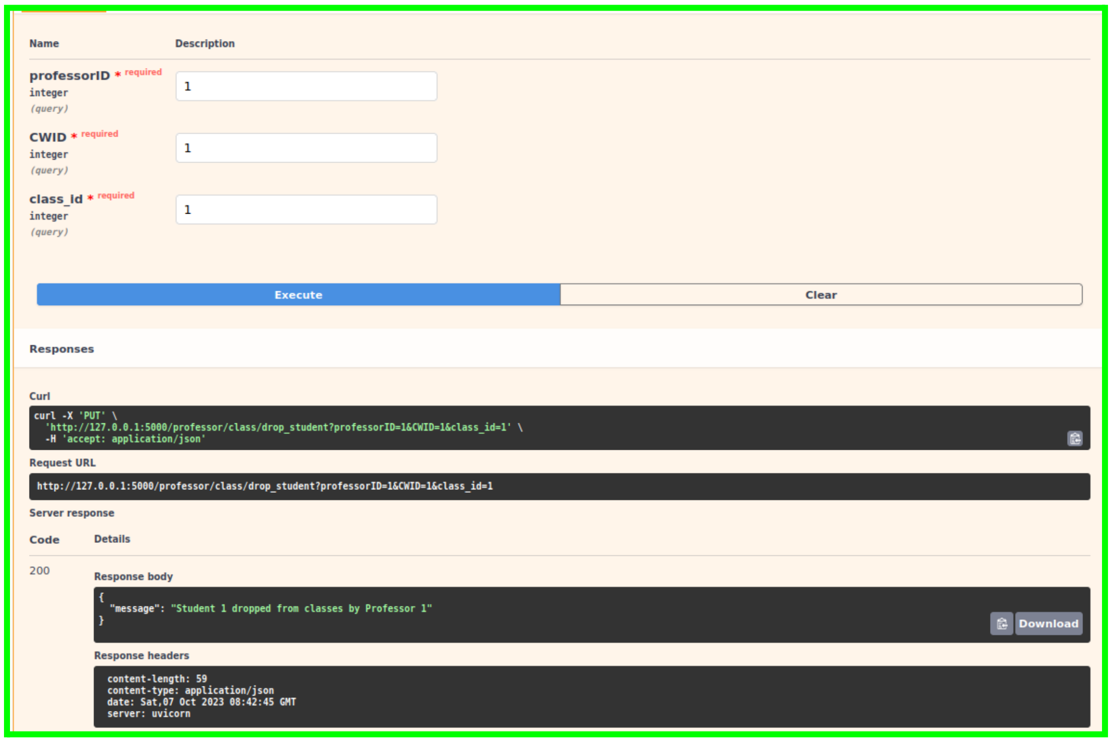
200 Response body

```
{
  "Class": [
    {
      "enrollmentID": 52,
      "CWID": 5,
      "classID": 2,
      "enrollmentDate": "2023-08-5",
      "dropped": 1
    },
    {
      "enrollmentID": 12,
      "CWID": 1,
      "classID": 2,
      "enrollmentDate": "your_enrollment_date",
      "dropped": 1
    }
  ]
}
```

Context: View the students that were dropped from the course

Result PUT Drop Student

- http://127.0.0.1:5000/professor/class/drop_student?professorID=1&CWID=1&class_id=1



Name	Description
professorID * required integer (query)	1
CWID * required integer (query)	1
class_id * required integer (query)	1

Curl

```
curl -X 'PUT' \
  'http://127.0.0.1:5000/professor/class/drop_student?professorID=1&CWID=1&class_id=1' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:5000/professor/class/drop_student?professorID=1&CWID=1&class_id=1
```

Server response

Code: 200

Response body

```
{
  "message": "Student 1 dropped from classes by Professor 1"
}
```

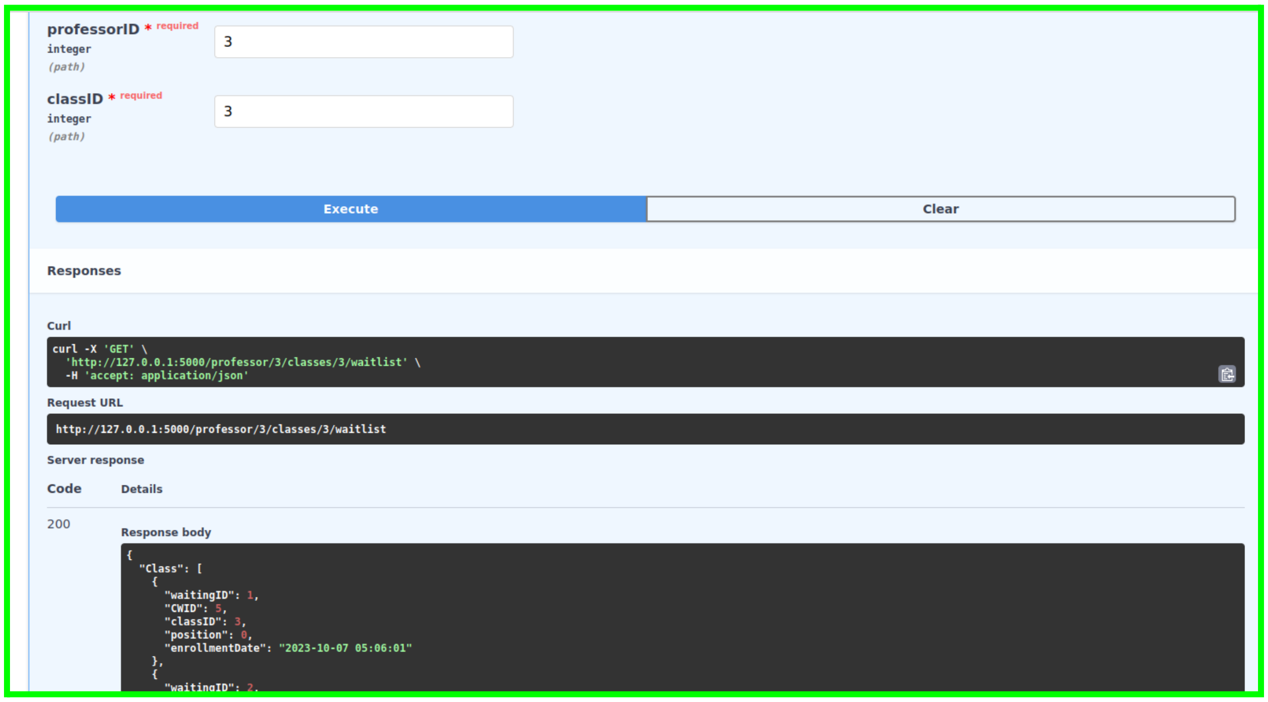
Response headers

```
content-length: 59
content-type: application/json
date: Sat,07 Oct 2023 08:42:45 GMT
server: uvicorn
```

Context: The professor is able to drop students administratively

Result GET Class Waitlist

- http://127.0.0.1:5000/professor/3/classes/3/waitlist



professorID * required
integer
(path)
3

classID * required
integer
(path)
3

Responses

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:5000/professor/3/classes/3/waitlist' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:5000/professor/3/classes/3/waitlist
```

Server response

Code Details

200

Response body

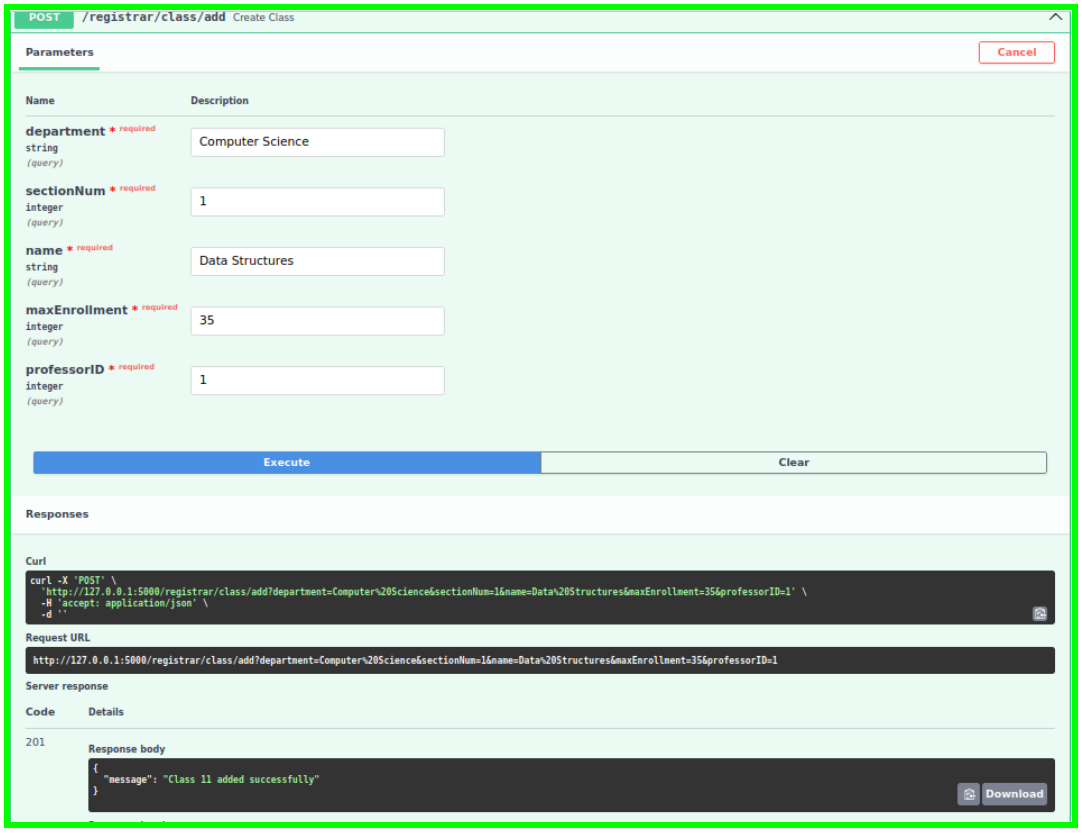
```
{
  "Class": [
    {
      "waitingID": 1,
      "CWID": 5,
      "classID": 3,
      "position": 0,
      "enrollmentDate": "2023-10-07 05:06:01"
    },
    {
      "waitingID": 2,
```

Context: The professor is able to view the waitlist positions for their classes.

Registrar

Result POST Add Class

- http://localhost:5000/registrar/class/add?department=Computer%20Science&sectionNum=1&name=Data%20Structures&maxEnrollement=35&currentEnrollment=21&professorID=1'


POST /registrar/class/add Create Class

Parameters

Name	Description
department * required string (query)	Computer Science
sectionNum * required integer (query)	1
name * required string (query)	Data Structures
maxEnrollment * required integer (query)	35
professorID * required integer (query)	1

Responses

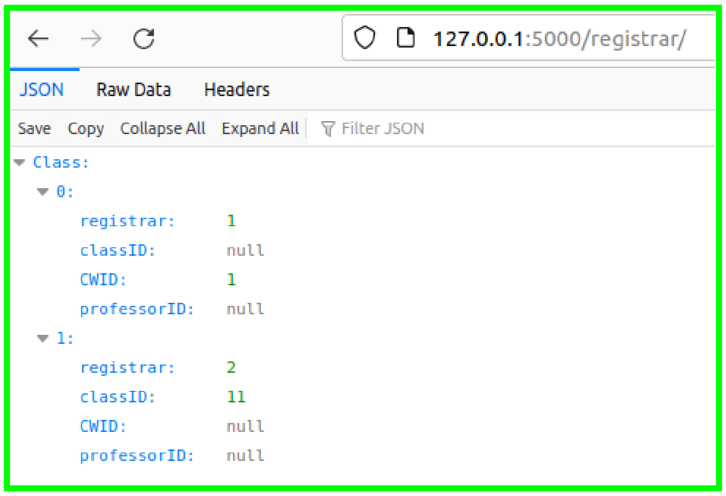
Curl
```
curl -X 'POST' \
  'http://127.0.0.1:5000/registrar/class/add?department=Computer%20Science&sectionNum=1&name=Data%20Structures&maxEnrollment=35&professorID=1' \
  -H 'accept: application/json' \
  -d ''
```

Request URL
```
http://127.0.0.1:5000/registrar/class/add?department=Computer%20Science&sectionNum=1&name=Data%20Structures&maxEnrollment=35&professorID=1
```

Server response

Code 201

Response body
```
{
  "message": "Class 11 added successfully"
}
```

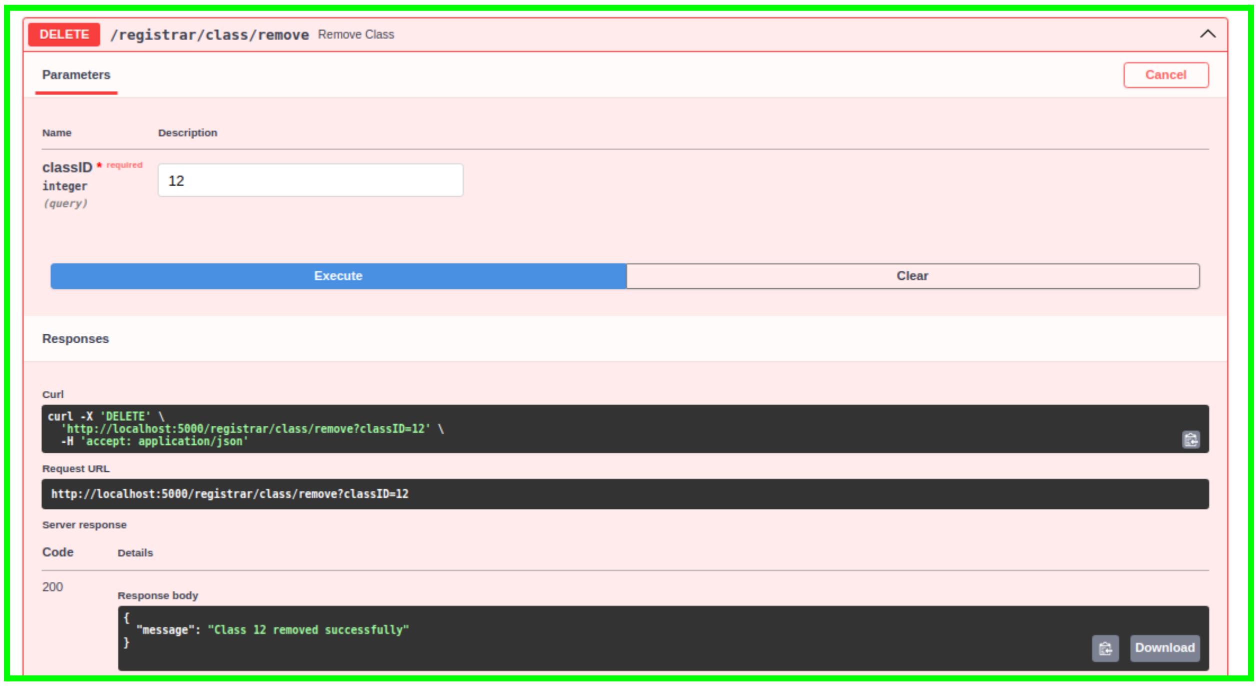

127.0.0.1:5000/registrar/

```
Class:
  0:
    registrar:    1
    classID:      null
    CWID:         1
    professorID:  null
  1:
    registrar:    2
    classID:      11
    CWID:         null
    professorID:  null
```

Context: This endpoint demonstrates the registrar's ability to add classes.

Result DELETE Remove Class

- http://localhost:5000/registrar/class/remove?classID=12



```
curl -X 'DELETE' \
  'http://localhost:5000/registrar/class/remove?classID=12' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:5000/registrar/class/remove?classID=12
```

Server response

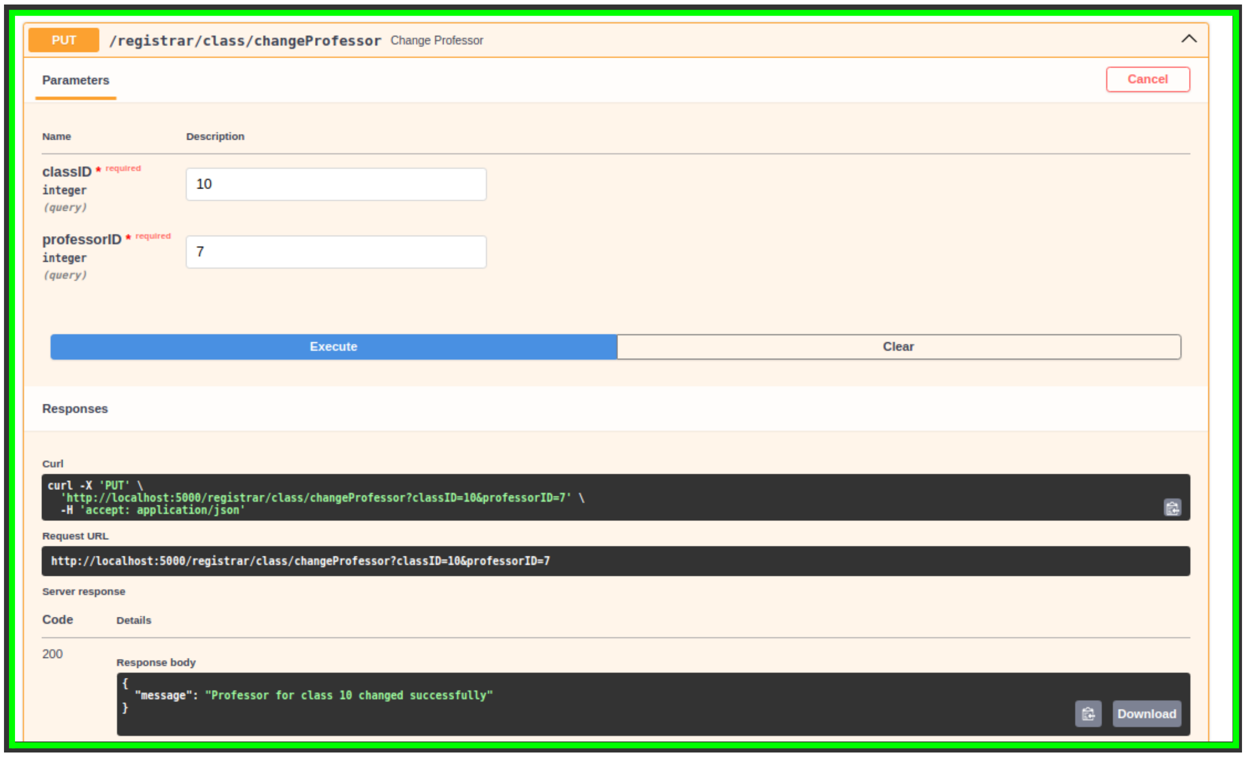
Code	Details
200	Response body

```
{
  "message": "Class 12 removed successfully"
}
```

Context: This endpoint shows the registrar’s ability to remove classes.

Result PUT Change Professor for a Class

- http://localhost:5000/registrar/class/changeProfessor?classID=10&professorID=7



classID * required
integer
(query)
10

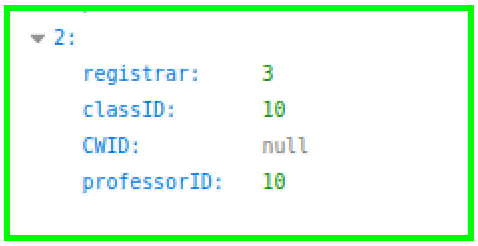
professorID * required
integer
(query)
7

```
curl -X 'PUT' \
  'http://localhost:5000/registrar/class/changeProfessor?classID=10&professorID=7' \
  -H 'accept: application/json'
```

```
http://localhost:5000/registrar/class/changeProfessor?classID=10&professorID=7
```

200

```
{
  "message": "Professor for class 10 changed successfully"
}
```

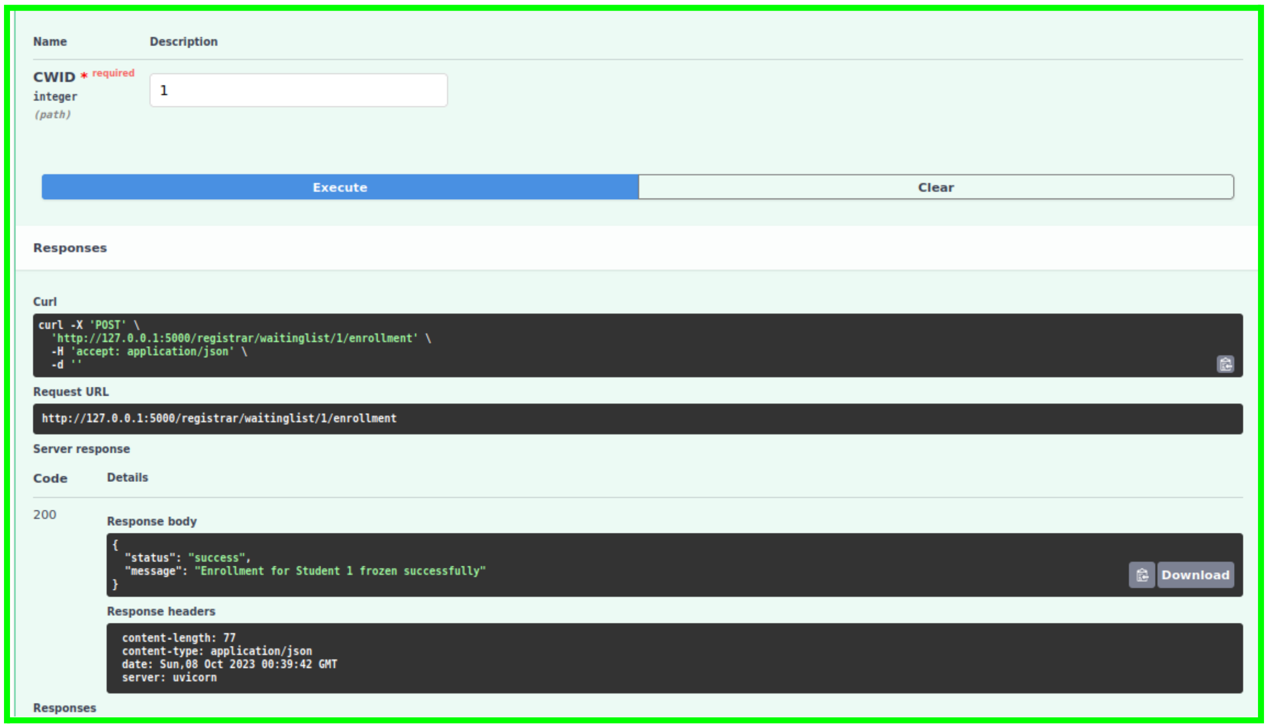


```
▾ 2:
    registrar:     3
    classID:       10
    CWID:          null
    professorID:   10
```

Context: This endpoint shows the registrar’s ability to change the professor teaching a specific class.

Result POST Freeze Enrollment

- http://127.0.0.1:5000/registrar/waitinglist/1/enrollment



CWID * required
integer
(path)
1

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:5000/registrar/waitinglist/1/enrollment' \
  -H 'accept: application/json' \
  -d ''
```

Request URL

```
http://127.0.0.1:5000/registrar/waitinglist/1/enrollment
```

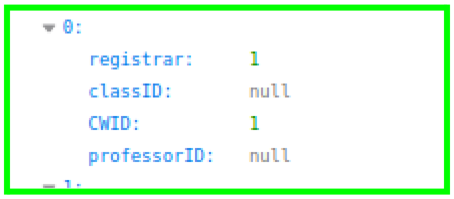
Server response

Code	Details
200	Response body

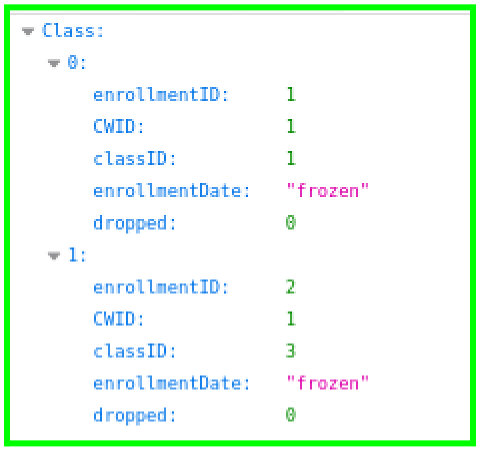
```
{
  "status": "success",
  "message": "Enrollment for Student 1 frozen successfully"
}
```

Response headers

```
content-length: 77
content-type: application/json
date: Sun,08 Oct 2023 00:39:42 GMT
server: uvicorn
```



```
▼ 0:
    registrar:     1
    classID:       null
    CWID:          1
    professorID:   null
```



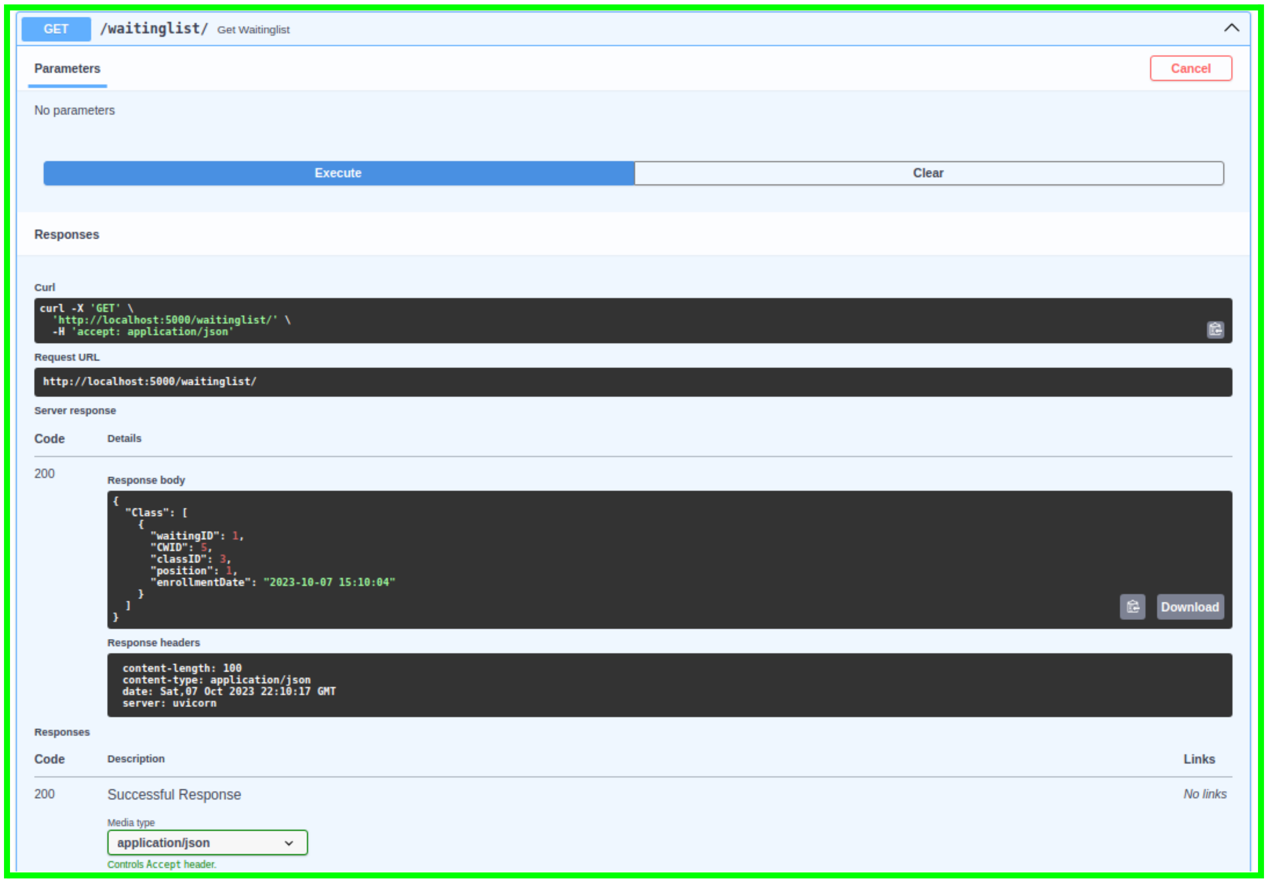
```
▼ Class:
  ▼ 0:
      enrollmentID:     1
      CWID:             1
      classID:          1
      enrollmentDate:   "frozen"
      dropped:          0
  ▼ 1:
      enrollmentID:     2
      CWID:             1
      classID:          3
      enrollmentDate:   "frozen"
      dropped:          0
```

Context: This endpoint freezes the enrollment for a student

Waitinglist

Result GET Waitlist

- http://localhost:5000/waitinglist/



Curl

```
curl -X 'GET' \
  'http://localhost:5000/waitinglist/' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:5000/waitinglist/
```

Server response

Code | Details

200

Response body

```
{
  "Class": [
    {
      "waitingID": 1,
      "CWID": 5,
      "classID": 3,
      "position": 1,
      "enrollmentDate": "2023-10-07 15:10:04"
    }
  ]
}
```

Response headers

```
content-length: 100
content-type: application/json
date: Sat,07 Oct 2023 22:10:17 GMT
server: uvicorn
```

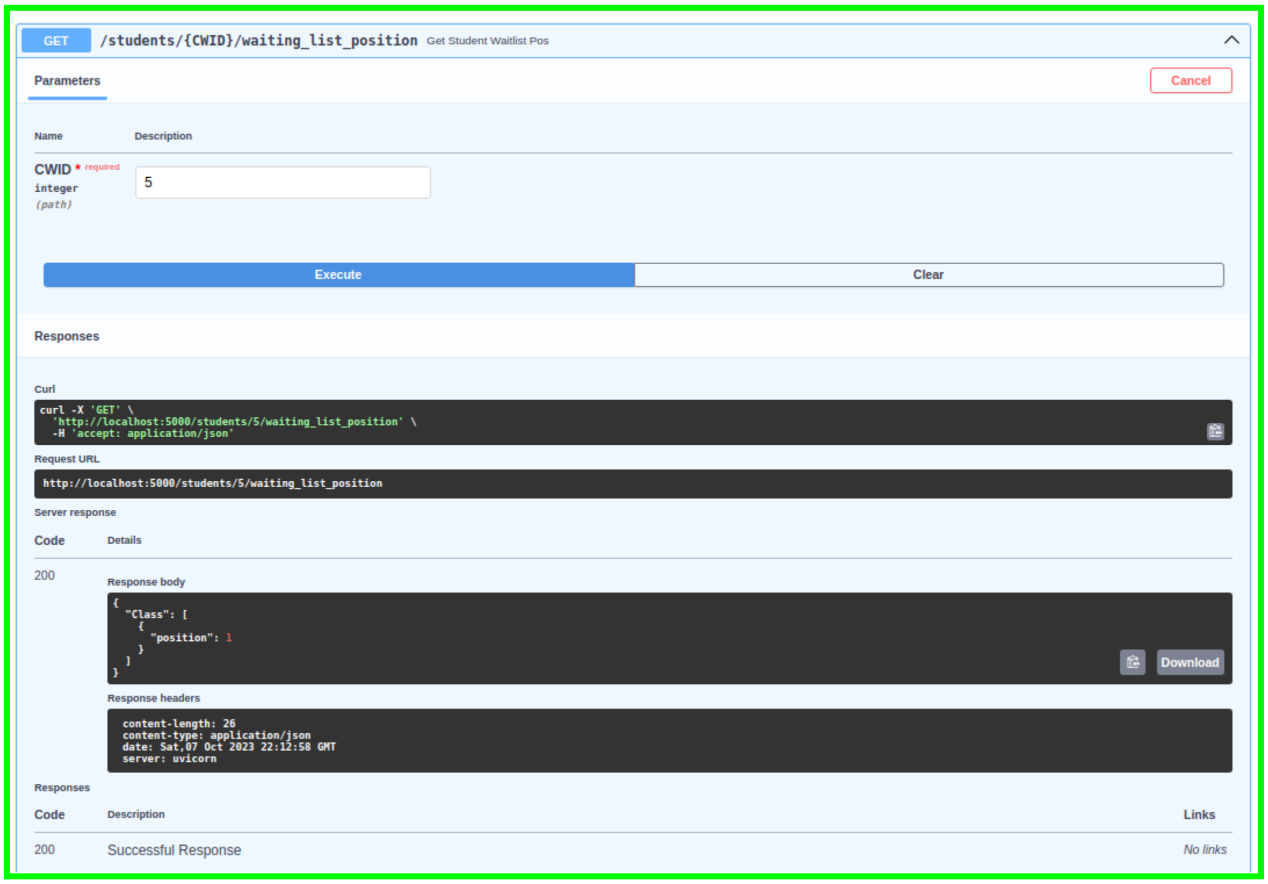
Responses

Code	Description	Links
200	Successful Response Media type: application/json Controls Accept header.	*No links*

Context: This endpoint shows the ability to display all the students that are currently waitlisted for any class.

Result GET Waitlist position of a Student

- http://localhost:5000/students/5/waiting_list_position



```
curl -X 'GET' \
  'http://localhost:5000/students/5/waiting_list_position' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:5000/students/5/waiting_list_position
```

Server response

Code	Details
200	Response body

```
{
  "Class": [
    {
      "position": 1
    }
  ]
}
```

Response headers

```
 content-length: 26
 content-type: application/json
 date: Sat,07 Oct 2023 22:12:58 GMT
 server: uvicorn
```

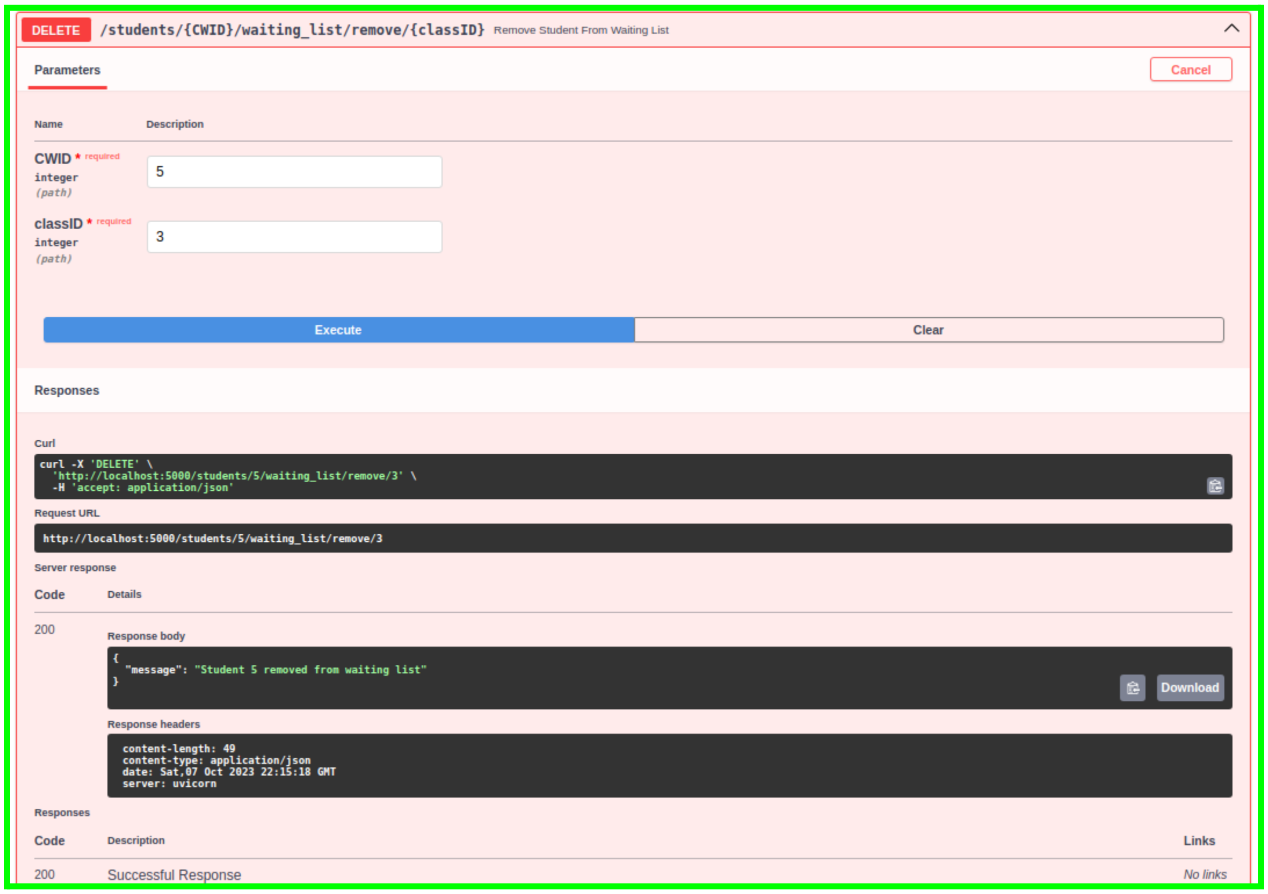
Responses

Code	Description	Links
200	Successful Response	No links

Context: Student 5 was able to view their position on the waitlist for the class they attempted to enroll in.

Result DELETE Remove Student for Waitlist

- http://localhost:5000/students/5/waiting_list/remove/3



Curl

```
curl -X 'DELETE' \
  'http://localhost:5000/students/5/waiting_list/remove/3' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:5000/students/5/waiting_list/remove/3
```

Server response

Code	Details
200	Response body

```
{
  "message": "Student 5 removed from waiting list"
}
```

Response headers

```
 content-length: 49
 content-type: application/json
 date: Sat,07 Oct 2023 22:15:18 GMT
 server: uvicorn
```

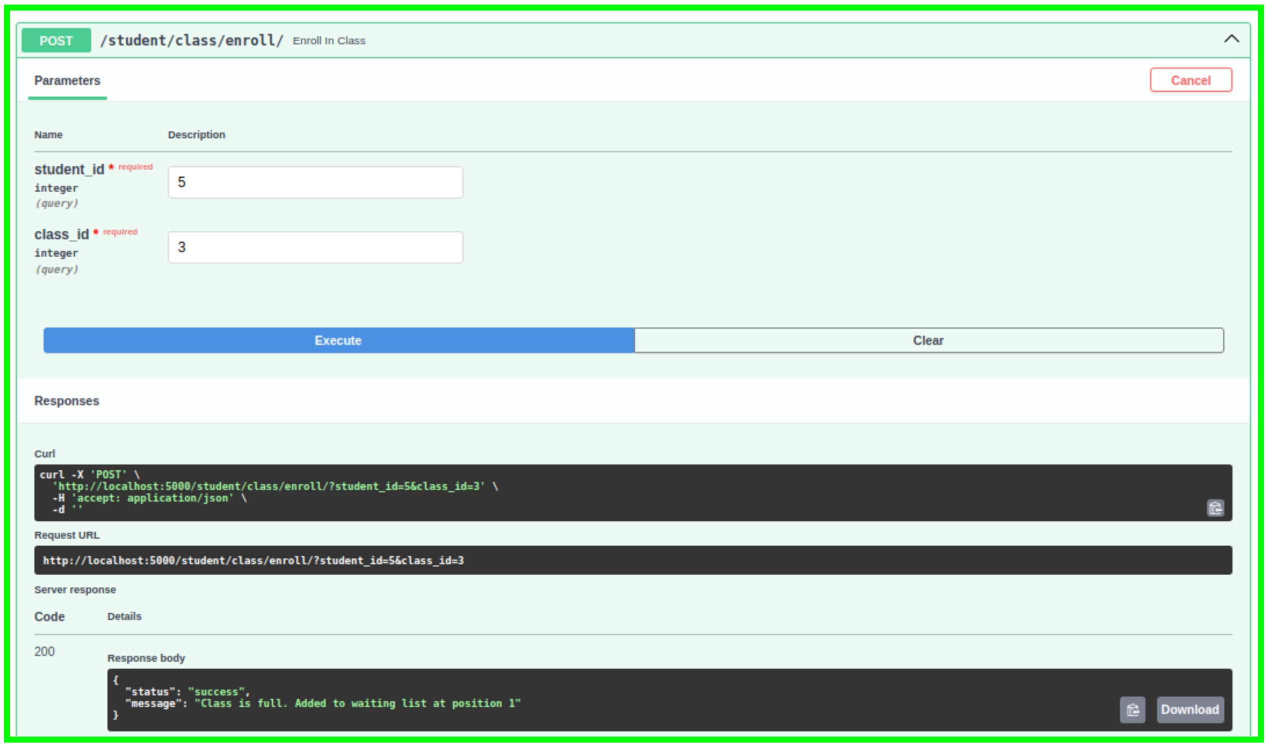
Responses

Code	Description	Links
200	Successful Response	*No links*

Context: Student 5 was able to get themselves removed from the waitlist for class with a classID of 3.

Result GET Waitlist for a Specific Professor

- http://localhost:5000/professor/3/classes/3/waitlist



Name	Description
student_id * required integer (query)	5
class_id * required integer (query)	3

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:5000/student/class/enroll/?student_id=5&class_id=3' \
  -H 'accept: application/json' \
  -d ''
```

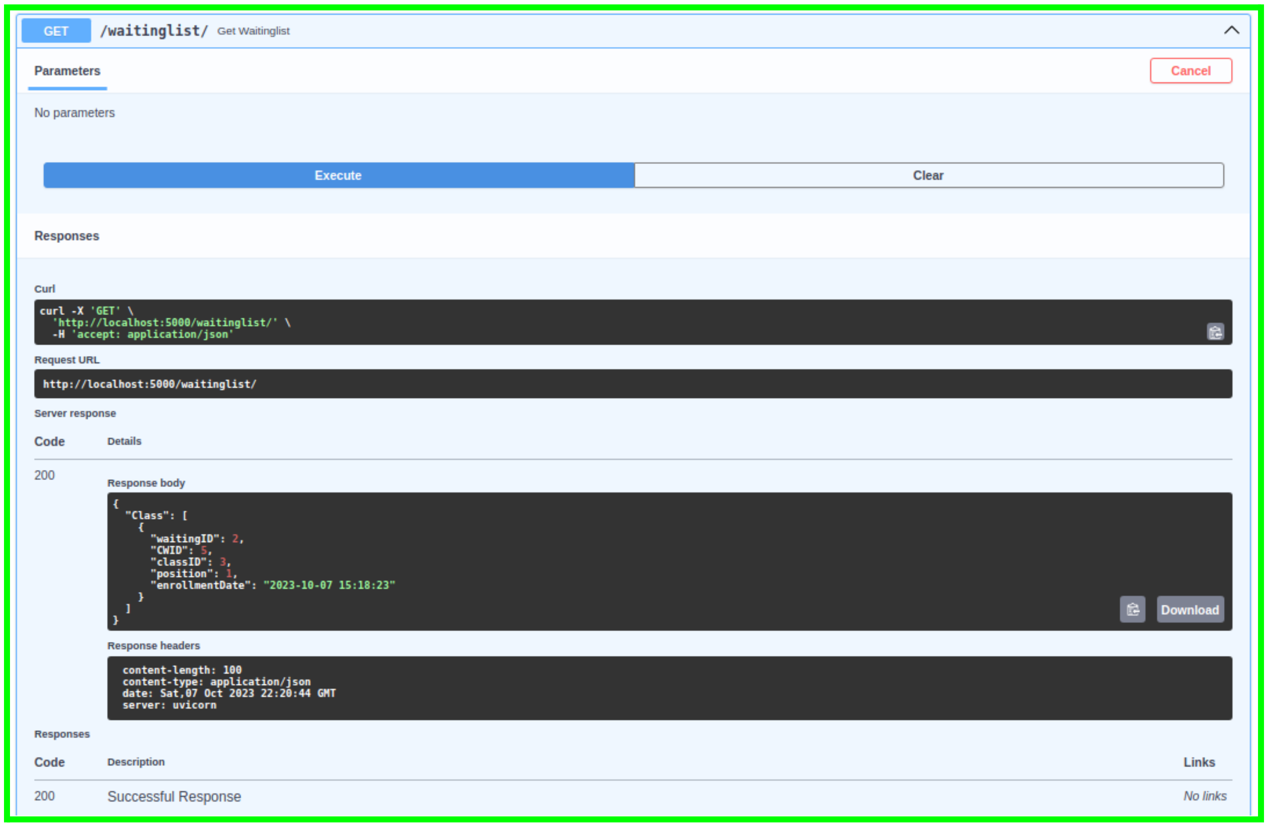
Request URL

```
http://localhost:5000/student/class/enroll/?student_id=5&class_id=3
```

Server response

Code	Details
200	Response body

```
{
  "status": "success",
  "message": "Class is full. Added to waiting list at position 1"
}
```



No parameters

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5000/waitinglist/' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:5000/waitinglist/
```

Server response

Code	Details
200	Response body

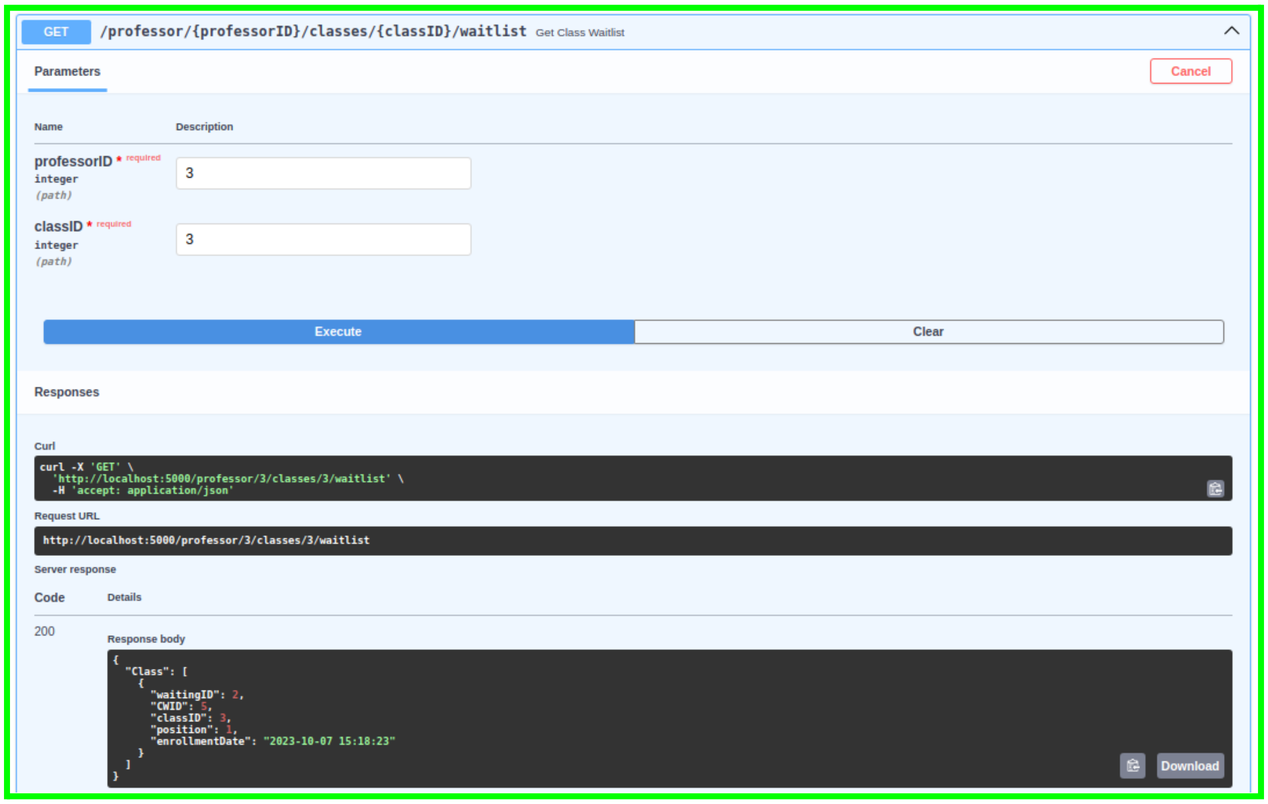
```
{
  "Class": [
    {
      "waitingID": 2,
      "CWID": 5,
      "classID": 3,
      "position": 1,
      "enrollmentDate": "2023-10-07 15:18:23"
    }
  ]
}
```

Response headers

```
 content-length: 100
 content-type: application/json
 date: Sat,07 Oct 2023 22:20:44 GMT
 server: uvicorn
```

Responses

Code	Description	Links
200	Successful Response	*No links*



```
/professor/{professorID}/classes/{classID}/waitlist
```

Name	Description
professorID * required integer (path)	3
classID * required integer (path)	3

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5000/professor/3/classes/3/waitlist' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:5000/professor/3/classes/3/waitlist
```

Server response

Code	Details
200	Response body

```
{
  "Class": [
    {
      "waitingID": 2,
      "CWID": 5,
      "classID": 3,
      "position": 1,
      "enrollmentDate": "2023-10-07 15:18:23"
    }
  ]
}
```

Context: To demonstrate this endpoint, student 5 was enrolled back into the waitlist for the class with a classID of 3 as shown in the first 2 screenshots. Then, the last screenshot shows how the professor was able to view the waitlist for that specific class.