



Velasquez, Gabrielle Ainsley L.

WD-302

## FINALS ACTIVITY 1: Node, Express, Mongo, & Vue

Please create the .env in the root directory with the following contents:

```
MONGODB_URI=mongodb+srv://wcserver2025:velasquez@cluster0.vbkgp7o.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0
```

### 1. Startup Commands

nodemon express.js

```
C:\Users\ainsh\Desktop\WC SERVER\BAKIT\hogwarts-vue-express-app>nodemon express.js
[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): ***!
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node express.js`
Hogwarts Portal running on port 5000
MongoDB connected successfully
```

npm run dev & npm run build

```
C:\Users\ainsh\Desktop\WC SERVER\hogwarts-vue-express-app\client>npm run build
> hogwarts-client@0.0.0 build
> vite build

vite v7.1.5 building for production...
✓ 42 modules transformed.
dist/index.html          0.43 kB  gzip: 0.28 kB
dist/assets/index-Cz9mizY9.css  9.47 kB  gzip: 2.64 kB
dist/assets/index-C45ATC4r.js  108.05 kB  gzip: 39.38 kB
✓ built in 1.33s

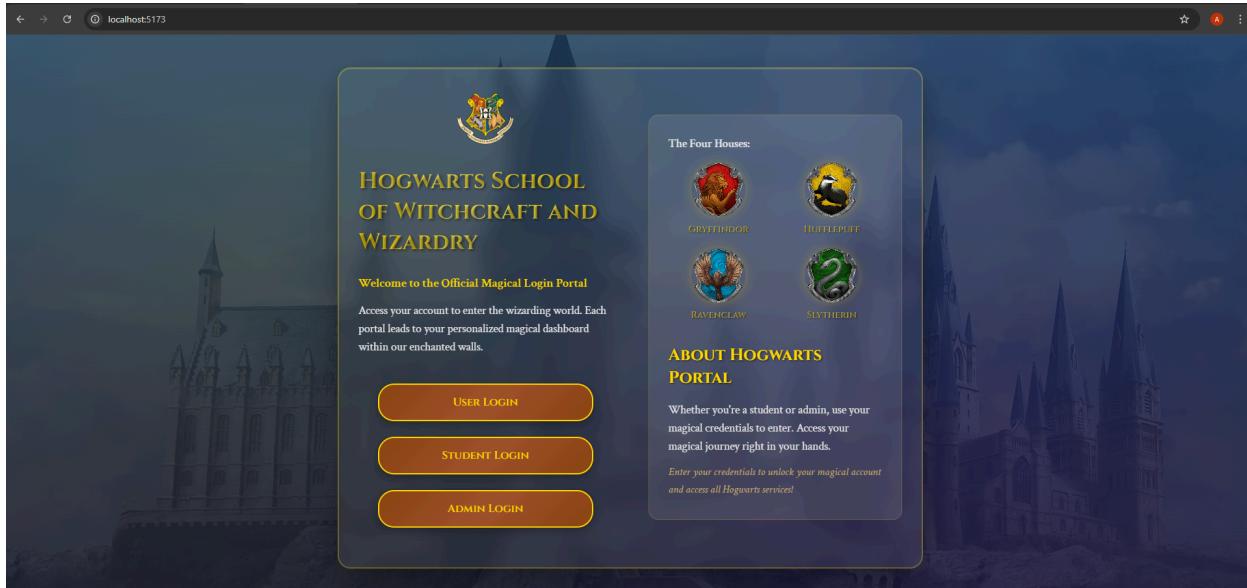
C:\Users\ainsh\Desktop\WC SERVER\hogwarts-vue-express-app\client>npm run dev
> hogwarts-client@0.0.0 dev
> vite

VITE v7.1.5 ready in 456 ms
→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

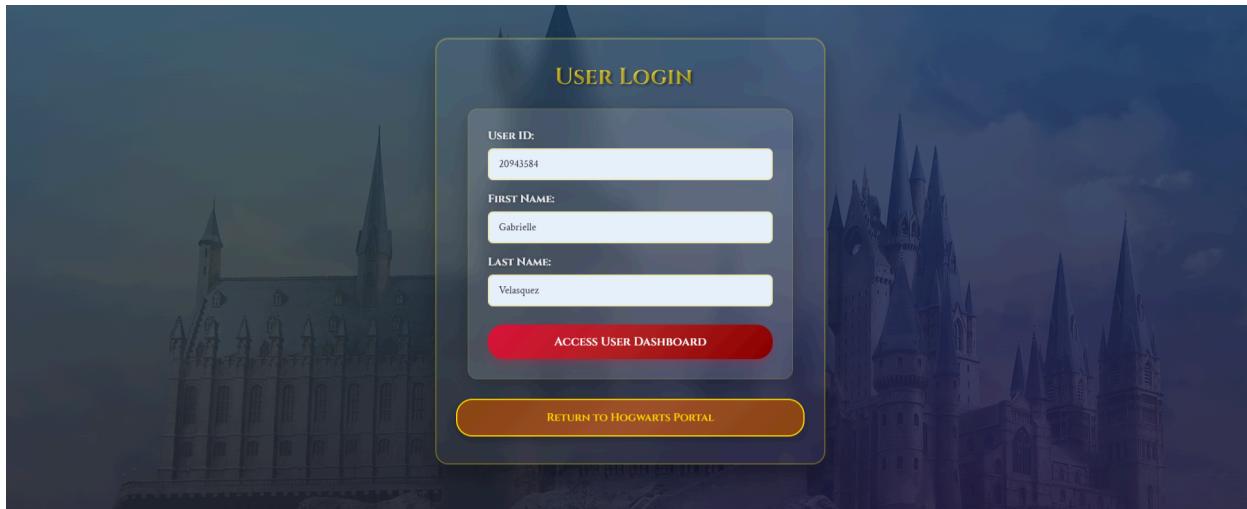


## 2. Application Output & API Testing

Home: localhost:5173



User Login: localhost:5173/user-form **POST /api/user-login**

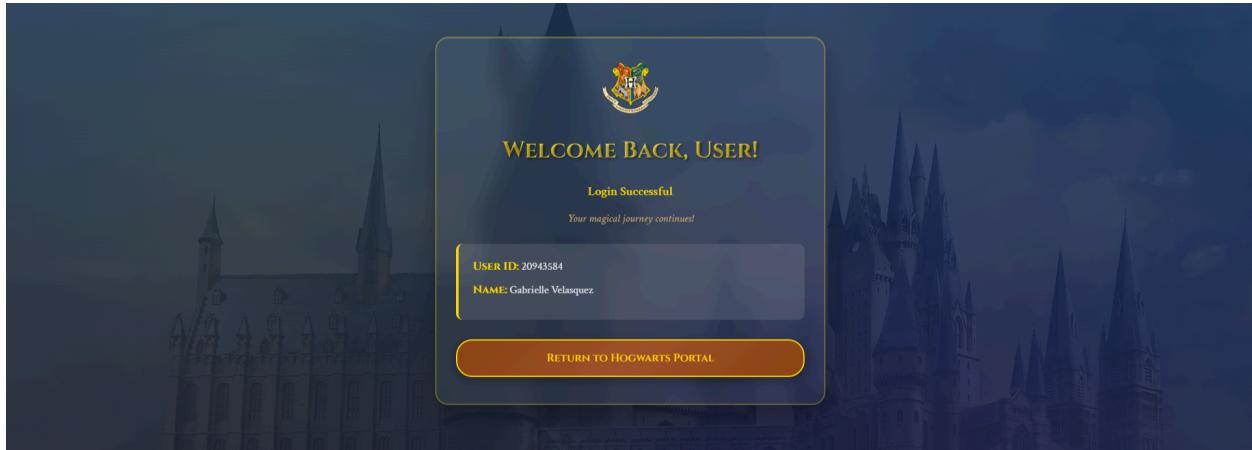




## HOLY ANGEL UNIVERSITY SCHOOL OF COMPUTING

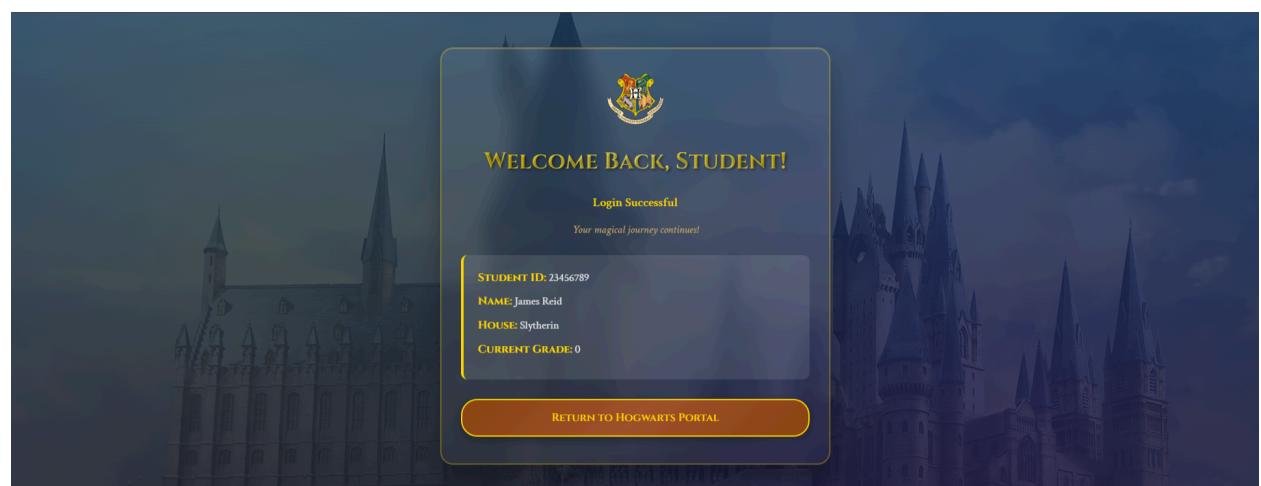
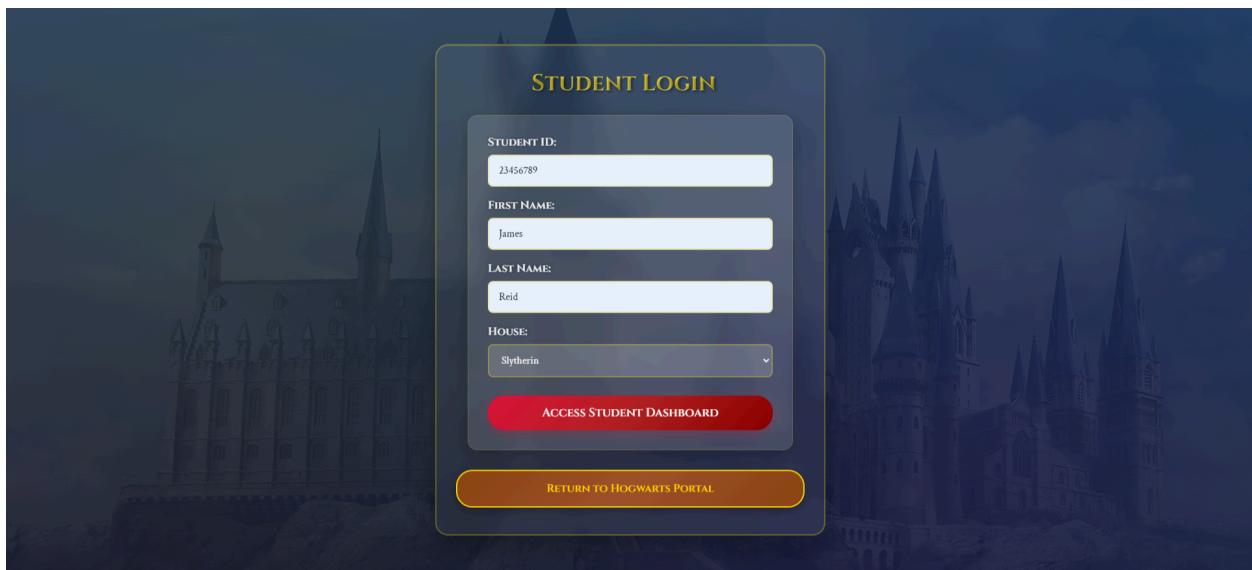


3



```
New User Created: {  
  userID: '20943584',  
  firstName: 'Gabrielle',  
  lastName: 'Velasquez',  
  _id: new ObjectId('68c9ff97cd68657727338425'),  
  __v: 0  
}
```

**Student Login:** localhost:5173/student-form **POST /api/student-login**





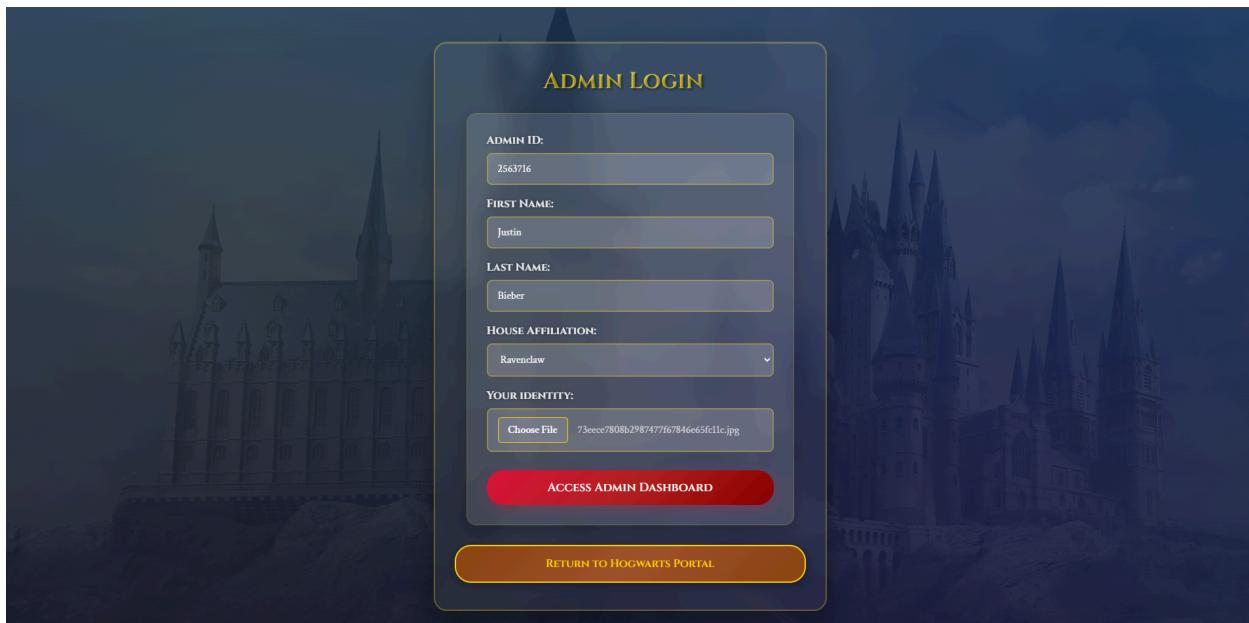
## HOLY ANGEL UNIVERSITY SCHOOL OF COMPUTING



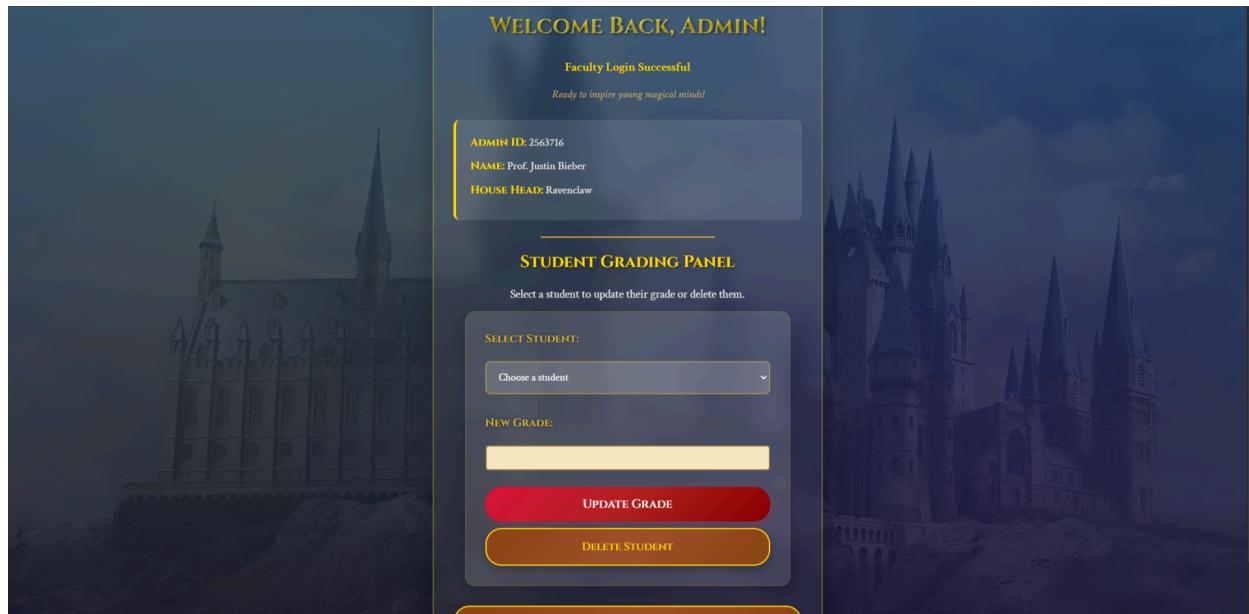
4

```
New Student Created: {  
  studentID: '23456789',  
  firstName: 'James',  
  lastName: 'Reid',  
  house: 'Slytherin',  
  grade: 0,  
  _id: new ObjectId('68c9ff46cd68657727338422'),  
  __v: 0  
}
```

**Admin Login:** localhost:5173/admin-form **POST /api/admin-login**



A screenshot of the Admin Login form. The background features a dark, atmospheric image of Hogwarts Castle. The form itself has a light blue gradient background and is titled "ADMIN LOGIN". It contains four input fields: "ADMIN ID" (2563716), "FIRST NAME" (Justin), "LAST NAME" (Bieber), and "HOUSE AFFILIATION" (Ravenclaw). Below these is a file upload field labeled "YOUR IDENTITY" with a placeholder "Choose File" and a file path "73ecc7808b2987477f67846e55fc11c.jpg". At the bottom are two buttons: a red "ACCESS ADMIN DASHBOARD" button and an orange "RETURN TO HOGWARTS PORTAL" button.



A screenshot of the Welcome Back, Admin! dashboard. The background is the same dark Hogwarts scene. The main header says "WELCOME BACK, ADMIN!". Below it, a message reads "Faculty Login Successful" and "Ready to inspire young magical minds!". A summary box shows "ADMIN ID: 2563716", "NAME: Prof. Justin Bieber", and "HOUSE HEAD: Ravenclaw". The central panel is titled "STUDENT GRADING PANEL" and includes a sub-header "Select a student to update their grade or delete them." It features a dropdown menu for "SELECT STUDENT" (Choose a student), a text input for "NEW GRADE", and two buttons: a red "UPDATE GRADE" button and an orange "DELETE STUDENT" button.



## HOLY ANGEL UNIVERSITY SCHOOL OF COMPUTING



5

**Update Student Grade:** `localhost:5173/user-result?status=success&type=admin`  
**POST /api/update-grade**

**STUDENT GRADING PANEL**

Select a student to update their grade or delete them.

**SELECT STUDENT:**  
James Reid (Grade: 0)

**NEW GRADE:**  
95

**UPDATE GRADE**

**DELETE STUDENT**

Grade updated successfully!

**STUDENT GRADING PANEL**

Select a student to update their grade or delete them.

**SELECT STUDENT:**  
Choose a student  
James Reid (Grade: 95)

**NEW GRADE:**

**UPDATE GRADE**

**DELETE STUDENT**

Grade updated successfully!

```
New Admin Created: {  
    adminID: '2563716',  
    firstName: 'Justin',  
    lastName: 'Bieber',  
    house: 'Ravenclaw',  
    profilePicturePath: 'client\\public\\uploads\\profilePicture-1758035539377-996722096.jpg'  
}  
Updated grade for James Reid to 95
```

**Delete Student:** `localhost:5173/user-result?status=success&type=admin`  
**DELETE /api/students/:studentID**

**STUDENT GRADING PANEL**

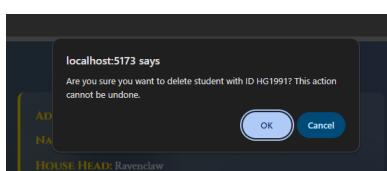
Select a student to update their grade or delete them.

**SELECT STUDENT:**  
Hermione Granger (Grade: 0)

Choose a student  
James Reid (Grade: 95)  
Hermione Granger (Grade: 0)

**UPDATE GRADE**

**DELETE STUDENT**



**STUDENT GRADING PANEL**

Select a student to update their grade or delete them.

**SELECT STUDENT:**  
Choose a student

**NEW GRADE:**

**UPDATE GRADE**

**DELETE STUDENT**

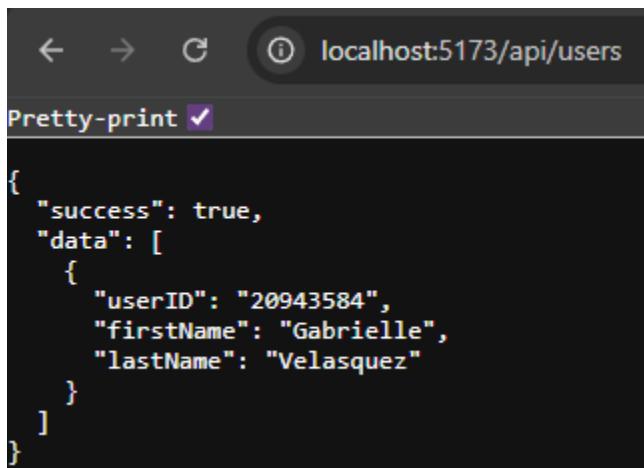
Student deleted successfully!



```
New Student Created: {  
  studentID: 'HG1991',  
  firstName: 'Hermione',  
  lastName: 'Granger',  
  house: 'Gryffindor',  
  grade: 0,  
  _id: new ObjectId('68ca0289cd68657727338436'),  
  __v: 0  
}  
Deleted student with ID: HG1991
```

## All API Endpoints

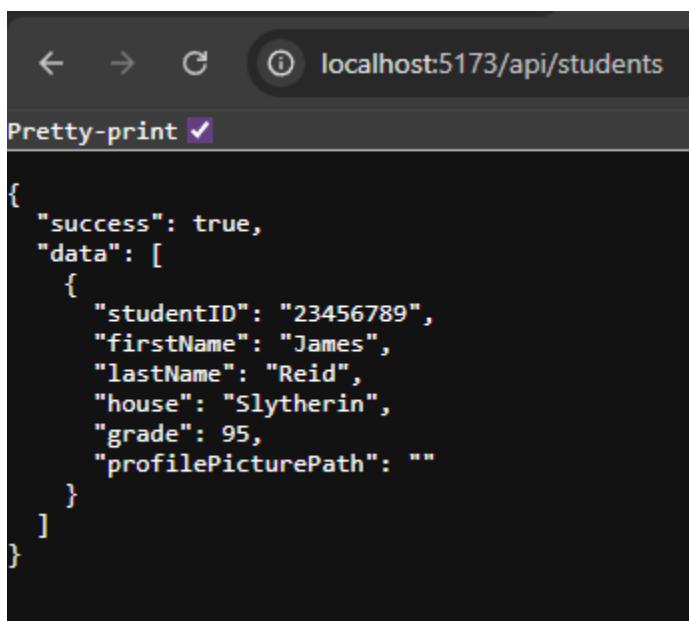
### GET /api/users



A screenshot of a browser window showing the JSON response for the GET /api/users endpoint. The URL in the address bar is "localhost:5173/api/users". The response is a success message with a single data item:

```
{  
  "success": true,  
  "data": [  
    {  
      "userID": "20943584",  
      "firstName": "Gabrielle",  
      "lastName": "Velasquez"  
    }  
  ]  
}
```

### GET /api/students



A screenshot of a browser window showing the JSON response for the GET /api/students endpoint. The URL in the address bar is "localhost:5173/api/students". The response is a success message with a single data item:

```
{  
  "success": true,  
  "data": [  
    {  
      "studentID": "23456789",  
      "firstName": "James",  
      "lastName": "Reid",  
      "house": "Slytherin",  
      "grade": 95,  
      "profilePicturePath": ""  
    }  
  ]  
}
```



## HOLY ANGEL UNIVERSITY SCHOOL OF COMPUTING



7

### GET /api/admins

A screenshot of a web browser window showing a JSON response. The URL in the address bar is "localhost:5173/api/admins". The response is a JSON object with "success": true and "data": [ { "adminID": "2563716", "firstName": "Justin", "lastName": "Bieber", "house": "Ravenclaw", "profilePicturePath": "client\\public\\uploads\\profilePicture-1758035539377-996722096.jpg" } ]

```
{
  "success": true,
  "data": [
    {
      "adminID": "2563716",
      "firstName": "Justin",
      "lastName": "Bieber",
      "house": "Ravenclaw",
      "profilePicturePath": "client\\public\\uploads\\profilePicture-1758035539377-996722096.jpg"
    }
  ]
}
```

### Error Page: localhost:5173/nonexistent

