Documentation

Important Utils
Following packages are extra installed while implementing backend functionality:

```
$ npm install bcrypt
$ npm install body-parser
```

Backend Documentation:
POST /login
Description: Use email and password to check whether this user is valid in the database. Return a JSON object with all the user information with a “check” field equals to 1 if succeed and return a JSON object with only a “check” field equals to 0 if failed.
Body Parameters:

- email: String
- password: String

Expected Response:
Successfully authenticated:
Return

```
{
  "check": 1,
  "_id": "633c847a87bbbf6f3bd1361c",
  "email": {
     "data": "caleb@123.com",
     "display": true
  },
  "password": "$2b$10$grXWK0rmF4WdriG7jVyu2OzXLwj1xdYtzJw8q/gkBQr18EY60O0ti",
  "name": {
     "data": "",
     "display": true
  },
  "dateofbirth": {
     "data": null,
     "display": true
  },
  "gender": {
     "data": "",
     "display": true
  },
  "Program": {
     "data": "",
     "display": true
  },
  "Description": {
    "data": "",
    "display": true
  }
}
```

Failed:

Return

```
{
  "check": 0
}
```

POST /signup

Description: adds a new student to the student collection in the database for newly registered users, return a json file which contains attribute result.

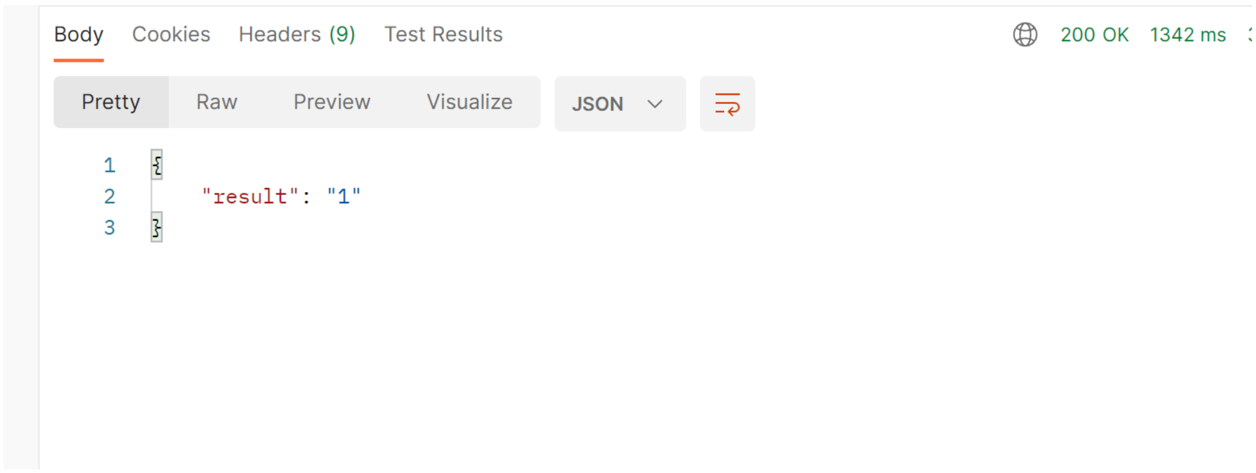
Body Parameters:

- Email: The email of the new registered student.
- Password: The password provided by the new student.

Expected Response:

- {result:"1"}

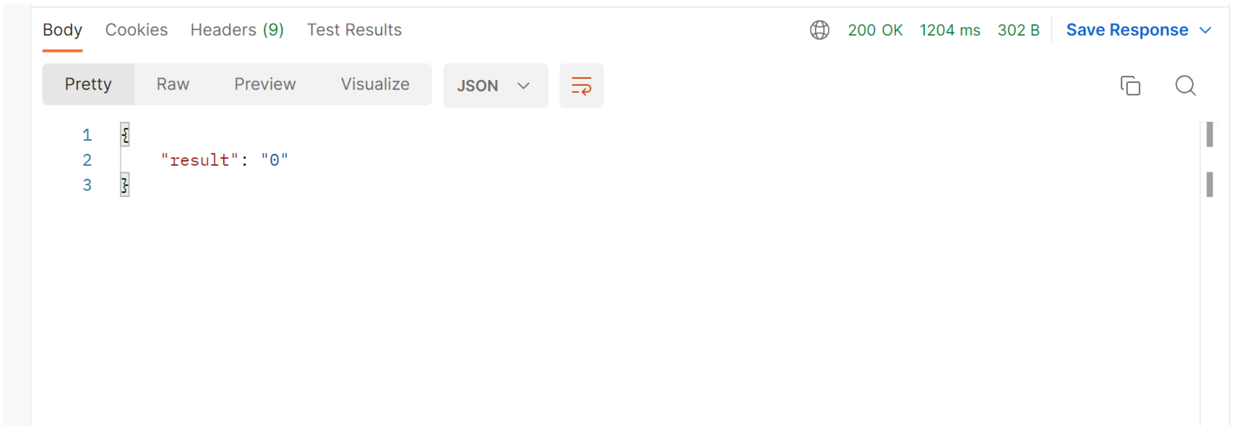
  The student has no duplicate email as another student in the database.



```
{
    "result": "1"
}
```

- {result:"0"}

  The student has no duplicate email as another student in the database or there is an error during the connection process.



```
{
    "result": "0"
}
```

DELETE /deleteUser

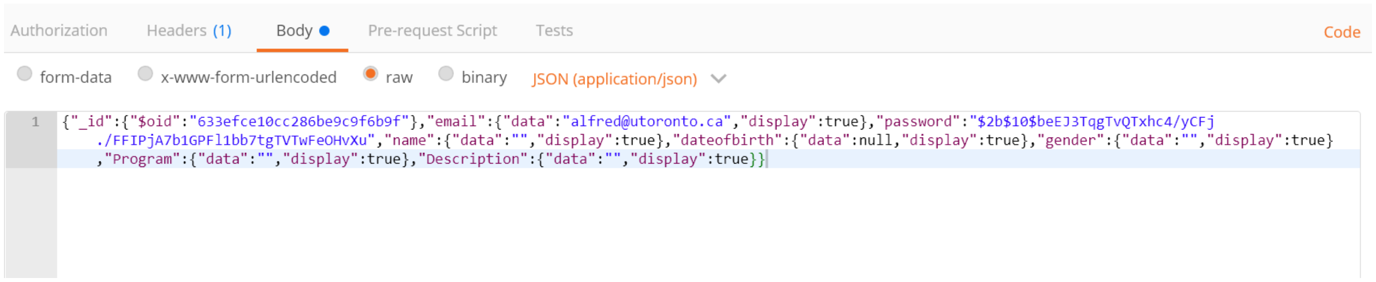
Description: Delete an existing user with the corresponding email passed by the body parameters from the student collection in the database, return a json file which contains 3 parts including the req (all body parameters), result of 0 (failure) /1 (success), and a message.

Body Parameters:

- Required
  - Email: The email of the existing student.
- Optional
  - All other information

Expected Response:

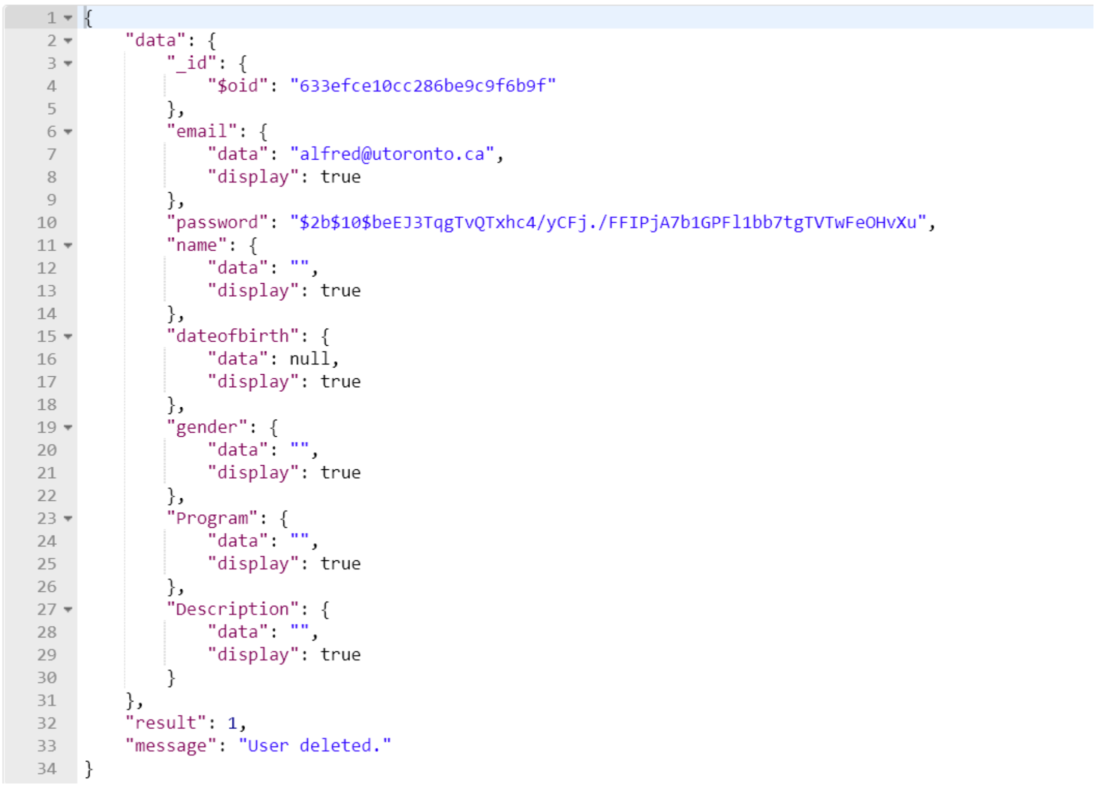
- Body parameter example:



```
{"_id":{"$oid":"633efce10cc286be9c9f6b9f"},"email":{"data":"alfred@utoronto.ca","display":true},"password":"$2b$10$beEJ3TqgTvQTxhc4/yCFj
./FFIPjA7b1GPFl1bb7tgTVTwFeOHvXu","name":{"data":"","display":true},"dateofbirth":{"data":null,"display":true},"gender":{"data":"","display":true}
,"Program":{"data":"","display":true},"Description":{"data":"","display":true}}
```

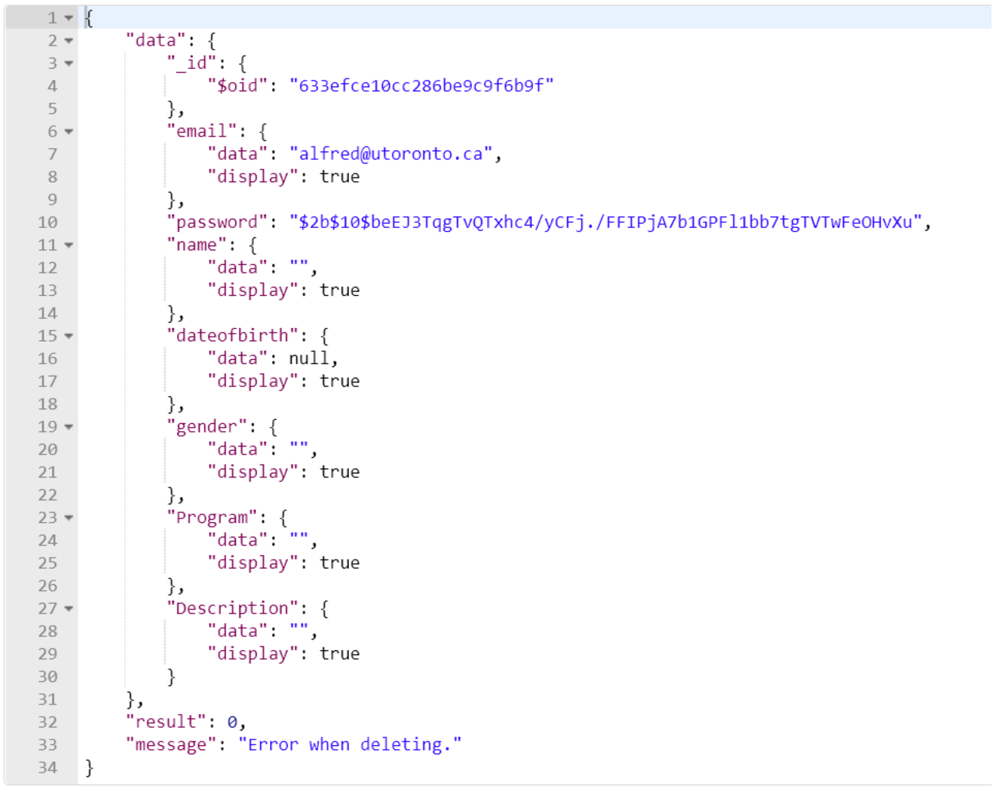
Note: Only the email part is needed to delete a user, but this is just an example with all the optional information

- On success (existing user with the given email is deleted in database):



```json
{
    "data": {
        "_id": {
            "$oid": "633efce10cc286be9c9f6b9f"
        },
        "email": {
            "data": "alfred@utoronto.ca",
            "display": true
        },
        "password": "$2b$10$beEJ3TqgTvQTxhc4/yCFj./FFIPjA7b1GPFl1bb7tgTVTwFeOHvXu",
        "name": {
            "data": "",
            "display": true
        },
        "dateofbirth": {
            "data": null,
            "display": true
        },
        "gender": {
            "data": "",
            "display": true
        },
        "Program": {
            "data": "",
            "display": true
        },
        "Description": {
            "data": "",
            "display": true
        }
    },
    "result": 1,
    "message": "User deleted."
}
```

- On failure



```json
{
    "data": {
        "_id": {
            "$oid": "633efce10cc286be9c9f6b9f"
        },
        "email": {
            "data": "alfred@utoronto.ca",
            "display": true
        },
        "password": "$2b$10$beEJ3TqgTvQTxhc4/yCFj./FFIPjA7b1GPFl1bb7tgTVTwFeOHvXu",
        "name": {
            "data": "",
            "display": true
        },
        "dateofbirth": {
            "data": null,
            "display": true
        },
        "gender": {
            "data": "",
            "display": true
        },
        "Program": {
            "data": "",
            "display": true
        },
        "Description": {
            "data": "",
            "display": true
        }
    },
    "result": 0,
    "message": "Error when deleting."
}
```

POST /edit

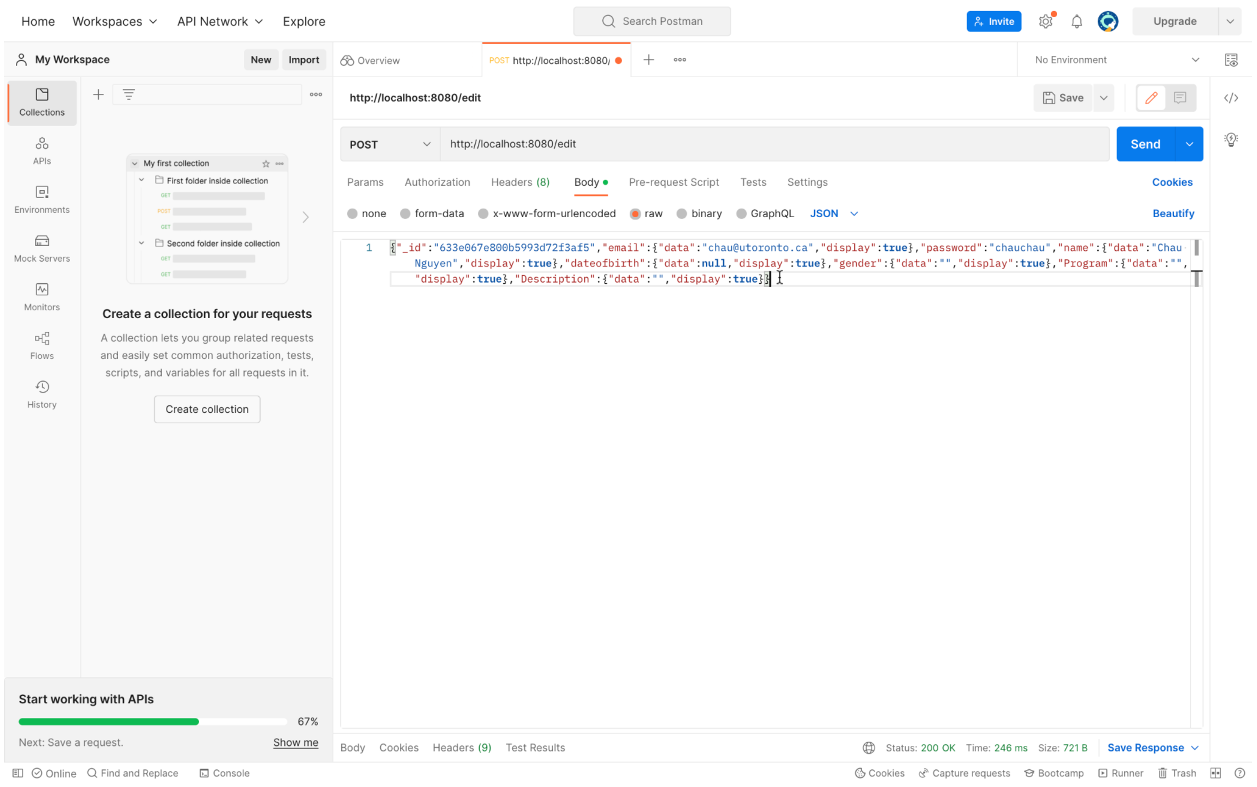
Description: Edit an existing user's profile with the corresponding email passed by the body parameters from the student collection in the database, return a json file which contains 3 parts including the req (all body parameters), result of 0 (failure) /1 (success), and a message.

Body Parameters:

- Required
  - Email: The email of the existing student.
  - Name, Gender, DateOfBirth, Program, Description: New data input by user.

Expected Response:

- Body parameter example:

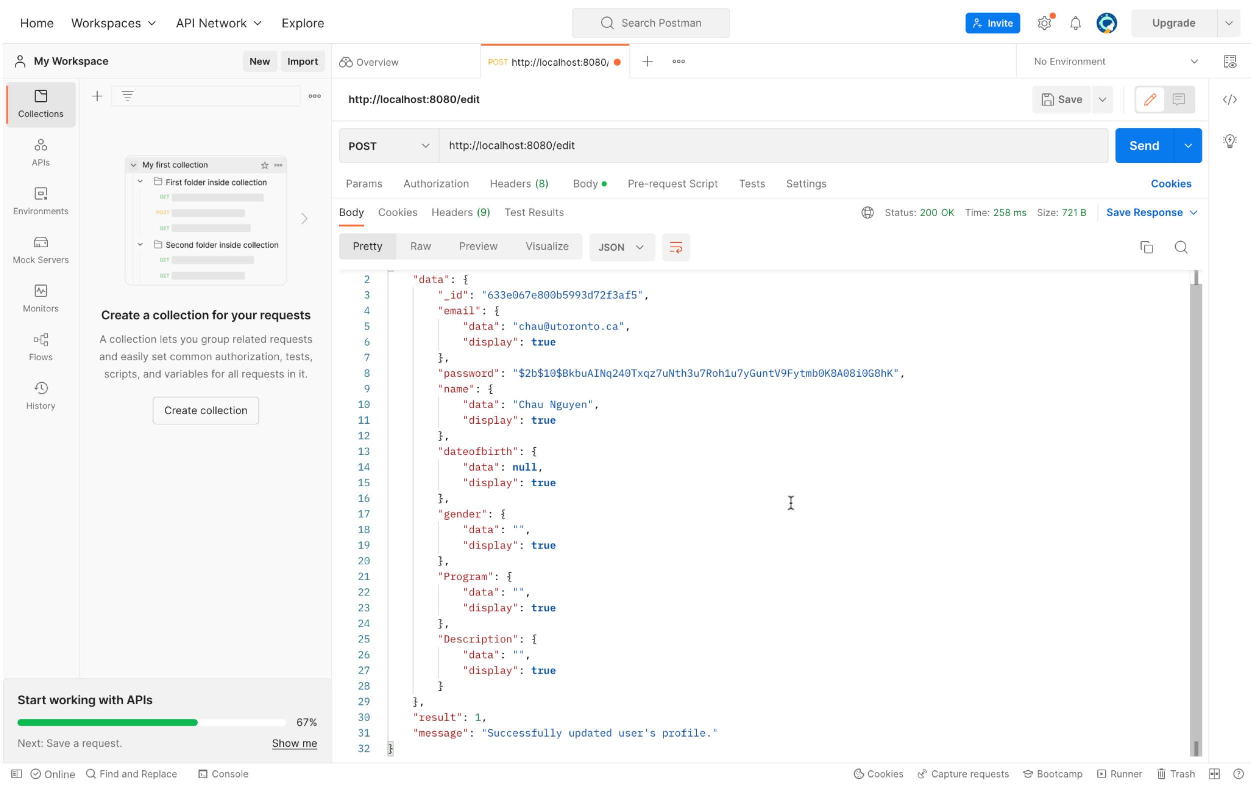


POST http://localhost:8080/edit

```
{"_id":"633e067e800b5993d72f3af5","email":{"data":"chau@utoronto.ca","display":true},"password":"chauchau","name":{"data":"Chau Nguyen","display":true},"dateofbirth":{"data":null,"display":true},"gender":{"data":"","display":true},"Program":{"data":"","display":true},"Description":{"data":"","display":true}}
```

Status: 200 OK Time: 246 ms Size: 721 B

- On success: Email address is registered in the database.

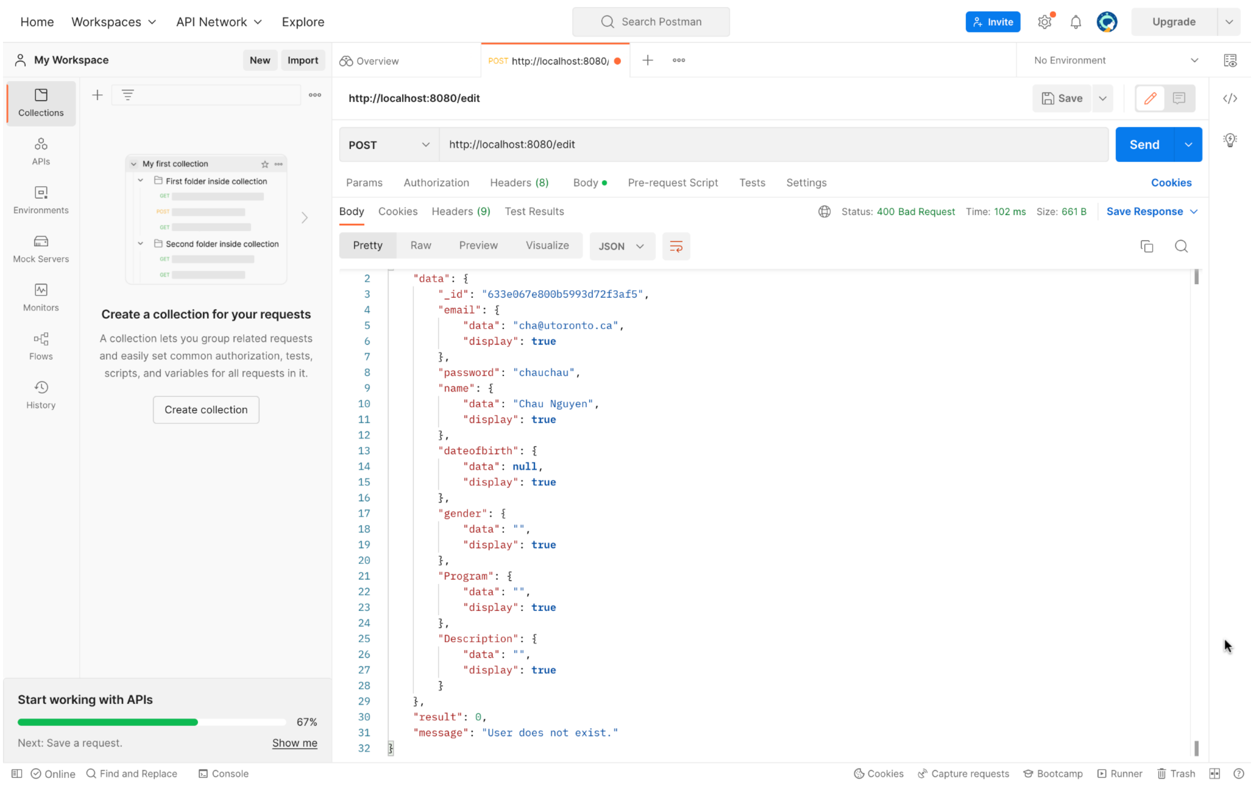


POST http://localhost:8080/edit

Status: 200 OK Time: 258 ms Size: 721 B

```
    "data": {
        "_id": "633e067e800b5993d72f3af5",
        "email": {
            "data": "chau@utoronto.ca",
            "display": true
        },
        "password": "$2b$10$BkbuAINq240Txqz7uNth3u7Roh1u7yGuntV9Fytmb0K8A08i0G8hK",
        "name": {
            "data": "Chau Nguyen",
            "display": true
        },
        "dateofbirth": {
            "data": null,
            "display": true
        },
        "gender": {
            "data": "",
            "display": true
        },
        "Program": {
            "data": "",
            "display": true
        },
        "Description": {
            "data": "",
            "display": true
        }
    },
    "result": 1,
    "message": "Successfully updated user's profile."
}
```

- On failure: Email address is not registered in the database. Connection to database/server fails.



POST http://localhost:8080/edit

Status: 400 Bad Request Time: 102 ms Size: 661 B

```
    "data": {
        "_id": "633e067e800b5993d72f3af5",
        "email": {
            "data": "cha@utoronto.ca",
            "display": true
        },
        "password": "chauchau",
        "name": {
            "data": "Chau Nguyen",
            "display": true
        },
        "dateofbirth": {
            "data": null,
            "display": true
        },
        "gender": {
            "data": "",
            "display": true
        },
        "Program": {
            "data": "",
            "display": true
        },
        "Description": {
            "data": "",
            "display": true
        }
    },
    "result": 0,
    "message": "User does not exist."
}
```

Frontend Documentation:

How to update React dependencies and start frontend:

```
$ npm install
$ npm start
```

File Structure

- The file /shortcut/frontend/src/routes.js clarifies all routes between pages.
- The folder /shortcut/frontend/src/Components contains frontend elements which are defined separately with respective .js and .css files.
- The folder /shortcut/frontend/src/Images contains image files that are/will be used in pages.
- The folder /shortcut/frontend/src/Pages contains folders for distinct pages. In the folders are .js and .css files for respective pages.

Page Summaries

- **Login**: This is the opening page of the website. An existing user can sign in with the correct email-password combination. The user will be sent to the Home page upon a successful login operation. A new user can navigate to the Signup page through a link.
- **Signup**: This page allows new users of the website to create a new account. The operation will be interrupted if the users provide invalid email/password inputs or fail to confirm the password. Users can go back to the Login page with a button.
- **Home**: This is the central page of the website. It allows users to navigate to the main sections of the website. (For sprint 1, users can only navigate to the Profile page.)
- **Profile**: This page allows users to view and edit their personal information. Users can choose to sign out or delete their accounts, after which they will be sent back to the Login page if the operation is successful. Users can also go back to the Home page through a button.
- Upcoming pages…

Important Utils

Following packages are extra installed while implementing frontend functionality:

```
$ npm install react-router-dom@6
$ npm install react-datepicker
$ npm install react-icons
```