API ENDPOINT

userController.js file :

* loginUser

Body : {email, password}

POST method

Res : status 202 USER

API: localhost:5000/api/users/login

* registerUser

Body: {(role, firstName, lastName, email. password,), <- required field, DOB, gender. Address.street, address.city, subjectID: []}

POST method

Res : status 201 json{"user id": userID,

            "email": email,

            "role": roles

}

API: localhost:5000/api/users

Since there are types of users, you can add more field into the body part. For example, enrolledChildren for parents or subjectID/parentID for students.

* changePass

Body: {email, oldPass, newPass}

PUT method

Res : status 200 json("Update successful")

API: localhost:5000/api/users/change-pass

* updateDetails

Body: {firstName, email, DOB, role}-> required field

PUT method

Res : status 202, update successful

The stated body attributes are the compulsory data to provides in the body. If data besides the compulsory

 const { role, firstName, lastName, email, DOB, gender, street, suburb, city, postCode, state, country } = req.body;

not provided in the body, it will remain the same as before. In addition, if wants to elective data (e.g: suburb) just put into the body “suburb” : “keiraville”. Email is un editable.

API: localhost:5000/api/users/update-details

* getUser

Params: /:field/:data

GET method

Res : status 201, all field of an user has except for password.

“field” is the name of the column in the db and data is the data to search.

e.g: [/email/asd@gmail.com](mailto:/email/asd@gmail.com) or /id/1006

NOTE: Please make sure to search only either by Id or email only ☺.

API : localhost:5000/api/users/find/:field/:data

* getUsers

Body: {}

GET method

Res: status 200, all users in each collections except admins (teacher, parent, students)

API : localhost:5000/api/users/users

* getChildren

Params: /parent’s id (/1006)

GET method

Res: status 200, all children of the parent.

API: localhost:5000/api/users/parent/children/:parent’s id

* enrollStudent

Body: {studentId, subjectId}

PUT method

Res: status 200, Student with id: ${id} enrolled into subject (${subjectId})

API: localhost:5000/api/users/teacher/enroll

* removeUser

Body: {userId}

DELETE method

Res:status 200, user removed

API:localhost:5000/api/users/remove