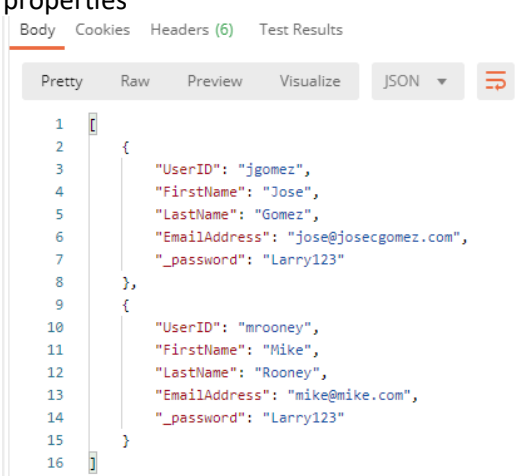
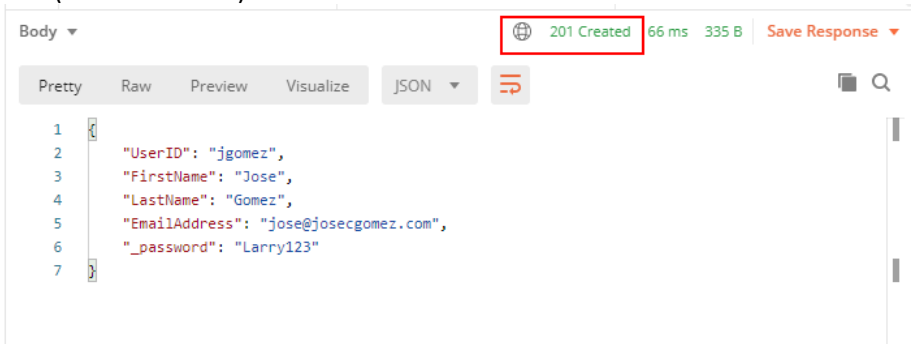
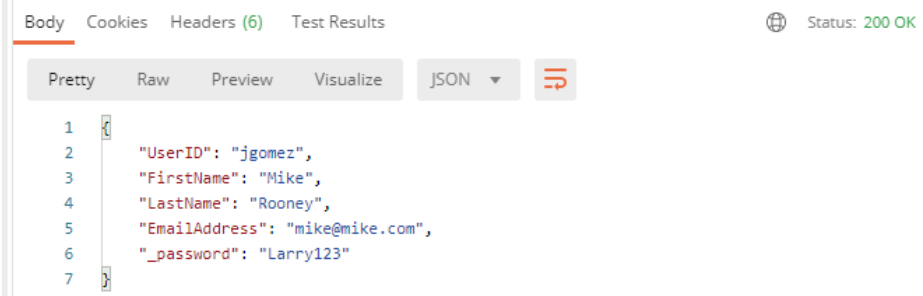
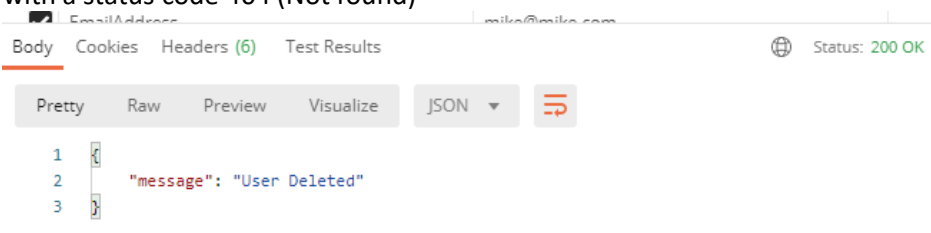

COP3503
Assignment 2

This assignment is an introduction to node and express and it will serve as the beginning back end for the next assignments it is imperative that this assignment be done and done correctly so that the next assignment after this can rely on it.

Write a Node application which will listen for request (web server) on port 3000 and it should provide the following routes and functionality for each route respectively.

Method	URL	Description
GET	/	<p>This is the default route the application should respond with a HELP Page in simple HTML that outlines the available functionality. A simple table which outlines each available request, the expected input and output data as well as any expected errors or requirements.</p> <p>Basically this description (with some examples) here are a few examples of what that page should look like (in terms of content)</p> <p>https://developer.twitter.com/en/docs/twitter-api/users/lookup/api-reference</p> <p>https://developer.okta.com/docs/reference/api/users/</p>
GET	/Users	<p>Should return a list of any user in the database (as a JSON array) with all their properties</p>  <pre>1 [2 { 3 "UserID": "jgomez", 4 "FirstName": "Jose", 5 "LastName": "Gomez", 6 "EmailAddress": "jose@josecgomez.com", 7 "_password": "Larry123" 8 }, 9 { 10 "UserID": "mrooney", 11 "FirstName": "Mike", 12 "LastName": "Rooney", 13 "EmailAddress": "mike@mike.com", 14 "_password": "Larry123" 15 } 16]</pre>
POST	/User	<p>Should Create a new User record and take in the following properties</p> <p>UserID Fist Name Last Name</p>

		<p>Email Address Password</p> <p>There should be error checking to ensure that the UserID doesn't already exist if it does an error should be returned If the user is successfully create it should return the user entity along with a 201 (Created Status)</p>  <pre> 1 { 2 "UserID": "jgomez", 3 "FirstName": "Jose", 4 "LastName": "Gomez", 5 "EmailAddress": "jose@josecgomez.com", 6 "_password": "Larry123" 7 } </pre>
GET	/User/{ID}	<p>Should return a specific User ID (the user ID provided in the URL) with all its properties if it exists. If the User ID is not found it should return an error 404 (Not Found) with an appropriate message</p>  <pre> 1 { 2 "statusCode": 404, 3 "message": "User not found" 4 } </pre> <pre> 1 { 2 "UserID": "jgomez", 3 "FirstName": "Jose", 4 "LastName": "Gomez", 5 "EmailAddress": "jose@josecgomez.com", 6 "_password": "Larry123" 7 } </pre>
PATCH	/User/{ID}	<p>Should allow you to Update (change) any data (except the ID) for an existing User. If the User isn't found return an error (same as above 404 not found)</p>  <pre> 1 { 2 "UserID": "jgomez", 3 "FirstName": "Mike", 4 "LastName": "Rooney", 5 "EmailAddress": "mike@mike.com", 6 "_password": "Larry123" 7 } </pre>

DELETE	/User/{ID}	<p>Should Delete a user (if it exists) if it doesn't exist it should return an Error with a status code 404 (Not found)</p> 
--------	------------	--

As extra credit provide an Input <HTML> form that can perform all of the above functions

Create User

Update User

Delete User

Get User Info

Make sure the user info is stored in a User Class (OOP!)

Any requests to other Routes not outlined above should redirect to the [GET] / Route which provides the Help

The assignment must be written using TypeScript, Node and Express no other external tools or libraries are necessary