

Exercises 2.10a

Installed CORS, bcrypt and express-validator via npm (git bash)

MINGW32/c:/Users/Jacqueline/Documents/GitHub/movie_api

```
Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ npm install cors
+ cors@2.8.5
added 2 packages from 2 contributors and audited 411 packages in 3.185s
```

```
1 package is looking for funding
  run 'npm fund' for details
```

```
found 0 vulnerabilities
```

```
Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ npm install bcrypt
```

```
> bcrypt@4.0.0 install C:\Users\Jacqueline\Documents\GitHub\movie_api\node_modules\bcrypt
> node-pre-gyp install --fallback-to-build
```

```
node-pre-gyp WARN Using needle for node-pre-gyp https download
[bcrypt] Success: "C:\Users\Jacqueline\Documents\GitHub\movie_api\node_modules\bcrypt\lib\binding\napi-v3\bcrypt_lib.node" is installed via remote
+ bcrypt@4.0.0
added 63 packages from 90 contributors and audited 510 packages in 15.614s
```

```
2 packages are looking for funding
  run 'npm fund' for details
```

```
found 0 vulnerabilities
```

```
Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ npm install @types/cors
+ @types/cors@2.8.6
added 1 package from 1 contributor and audited 524 packages in 3.086s
```

```
2 packages are looking for funding
  run 'npm fund' for details
```

```
found 0 vulnerabilities
```

```
Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ npm install @types/bcrypt
+ @types/bcrypt@3.0.0
added 1 package from 3 contributors and audited 525 packages in 8s
```

```
2 packages are looking for funding
  run 'npm fund' for details
```

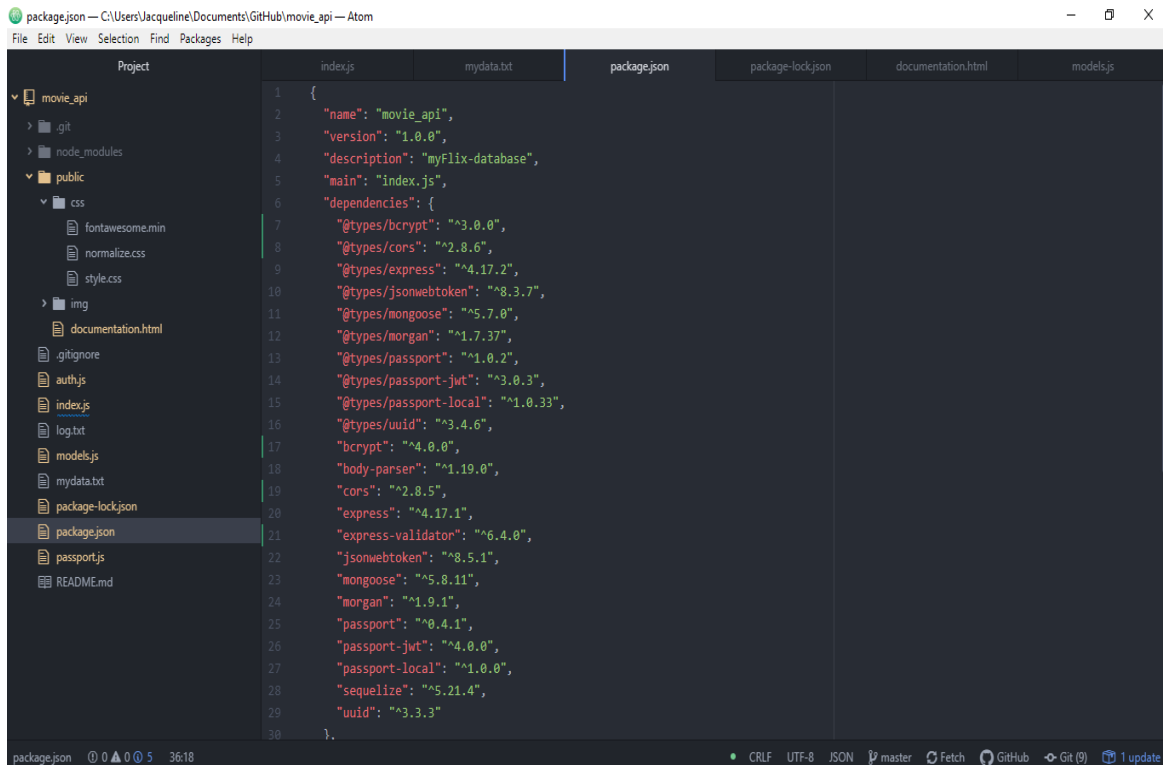
```
found 0 vulnerabilities
```

```
Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ npm install express-validator
+ express-validator@6.4.0
added 2 packages from 5 contributors and audited 528 packages in 4.684s

2 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities
```

Updated package.json:

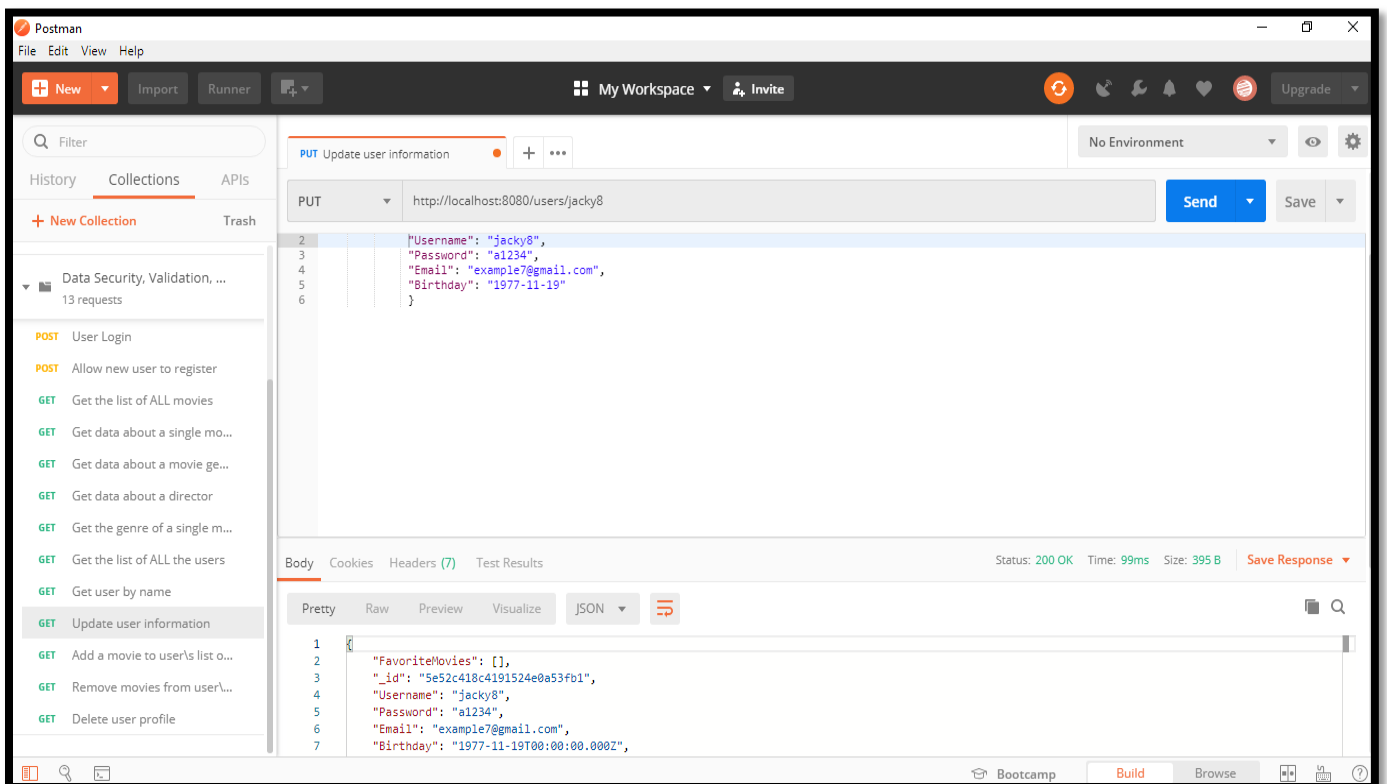


The screenshot shows the Atom editor interface with the 'package.json' file open. The file contains the following JSON structure:

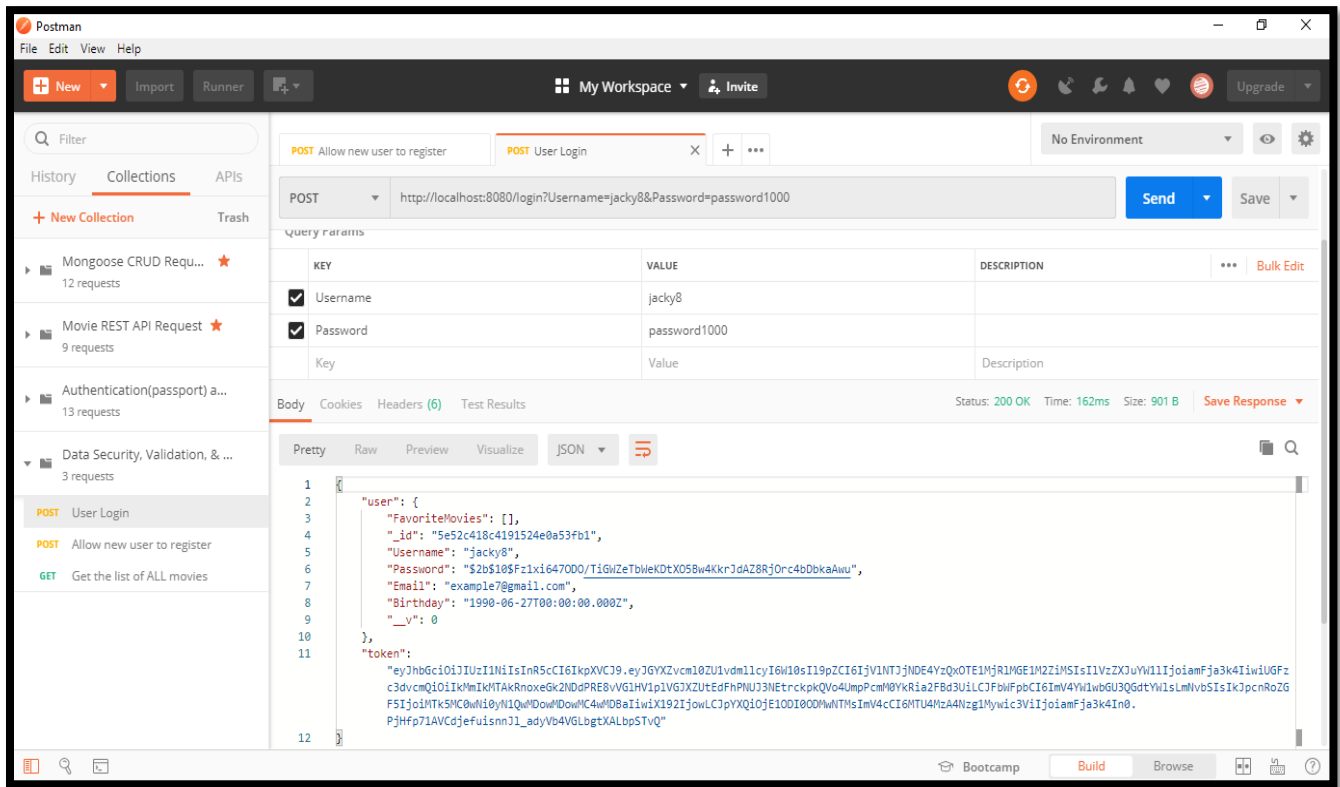
```
{
  "name": "movie_api",
  "version": "1.0.0",
  "description": "myFlix-database",
  "main": "index.js",
  "dependencies": {
    "@types/bcrypt": "^3.0.0",
    "@types/cors": "^2.8.6",
    "@types/express": "^4.17.2",
    "@types/jsonwebtoken": "^8.3.7",
    "@types/mongoose": "^5.7.0",
    "@types/morgan": "^1.7.37",
    "@types/passport": "^1.0.2",
    "@types/passport-jwt": "^3.0.3",
    "@types/passport-local": "^1.0.33",
    "@types/uuid": "^3.4.6",
    "bcrypt": "^4.0.0",
    "body-parser": "^1.19.0",
    "cors": "^2.8.5",
    "express": "^4.17.1",
    "express-validator": "^6.4.0",
    "jsonwebtoken": "^8.5.1",
    "mongoose": "^5.8.11",
    "morgan": "^1.9.1",
    "passport": "^0.4.1",
    "passport-jwt": "^4.0.0",
    "passport-local": "^1.0.0",
    "sequelize": "^5.21.4",
    "uuid": "^3.3.3"
  }
}
```

Postman screenshots (testing code before deploying via Heroku and MongoDB Atlas)

New User Registration

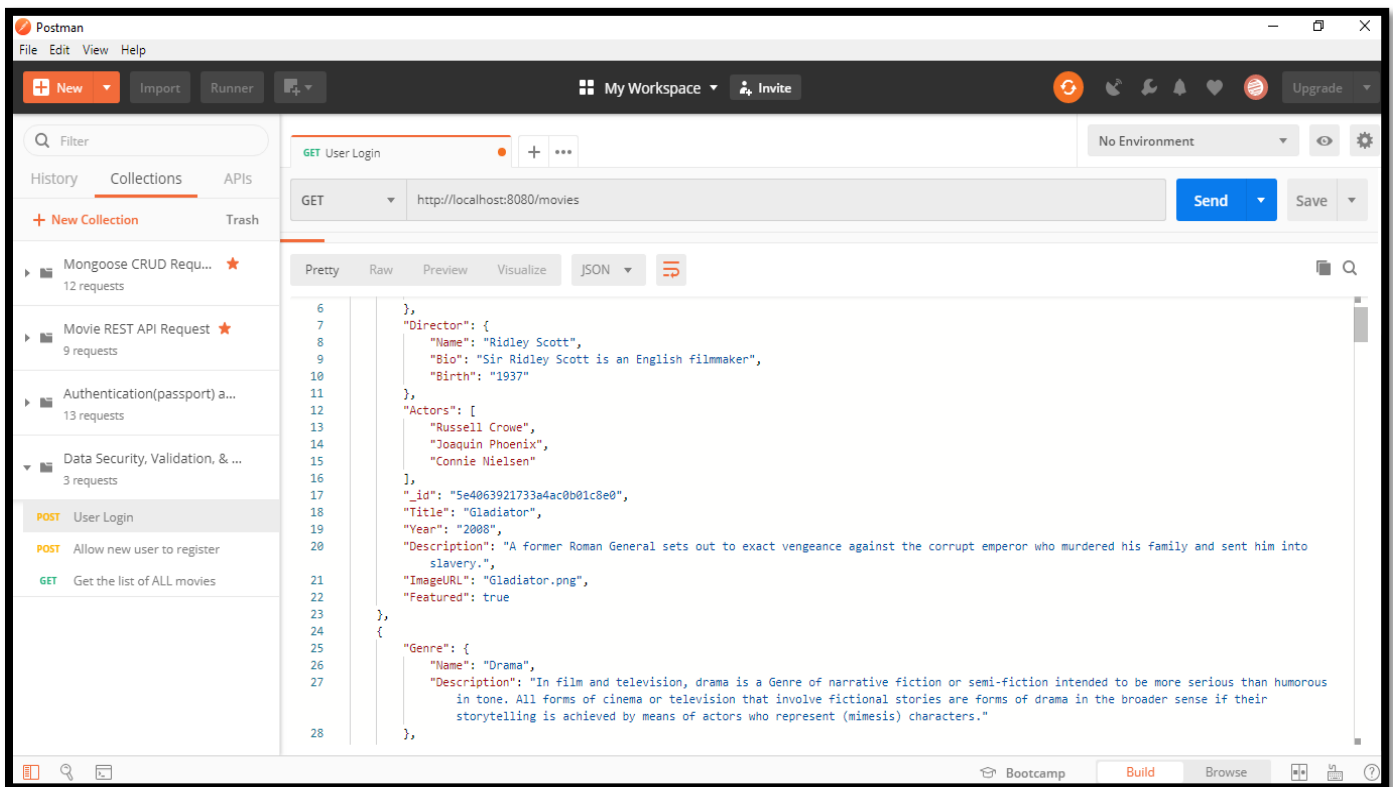


New User login and authorization token

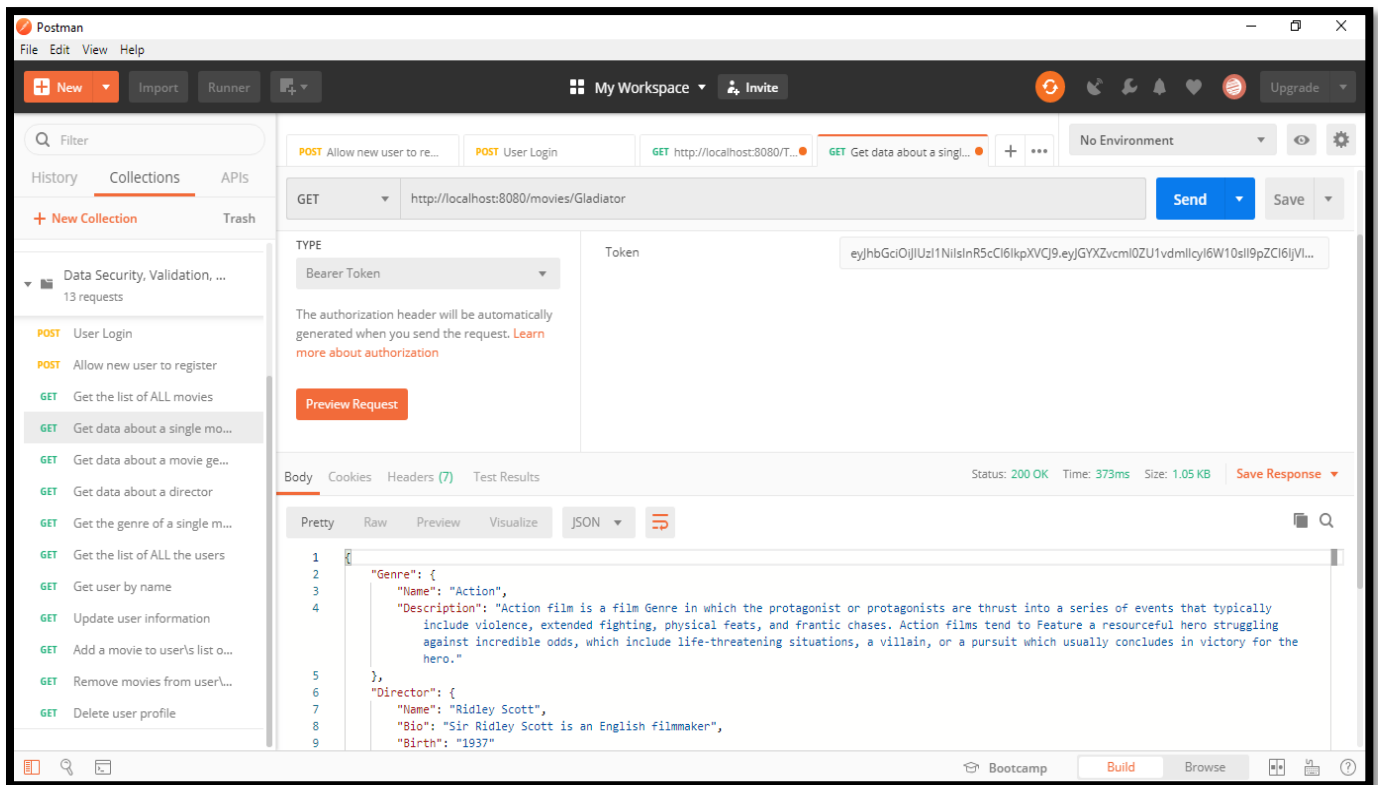


Postman screenshots with authorization token (11 requests)

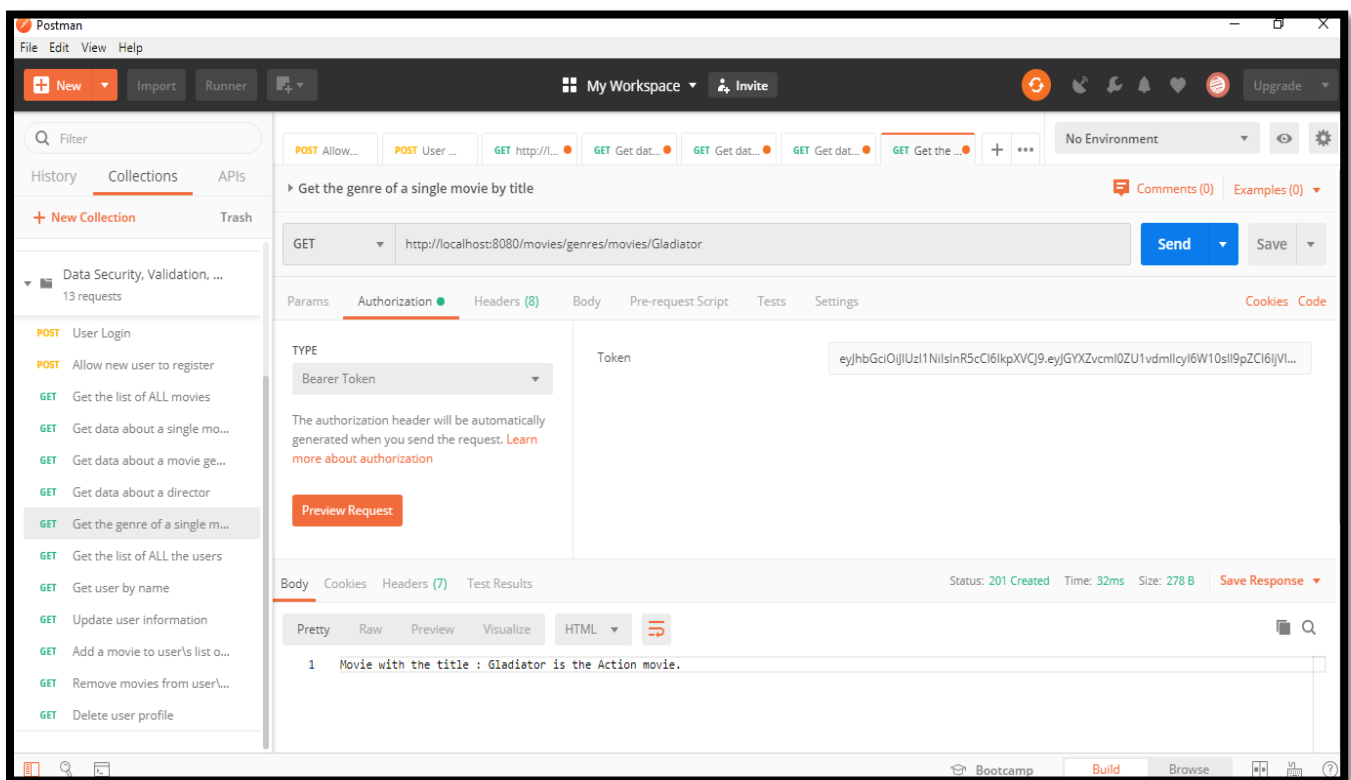
One: Return a list of ALL movies to the user:



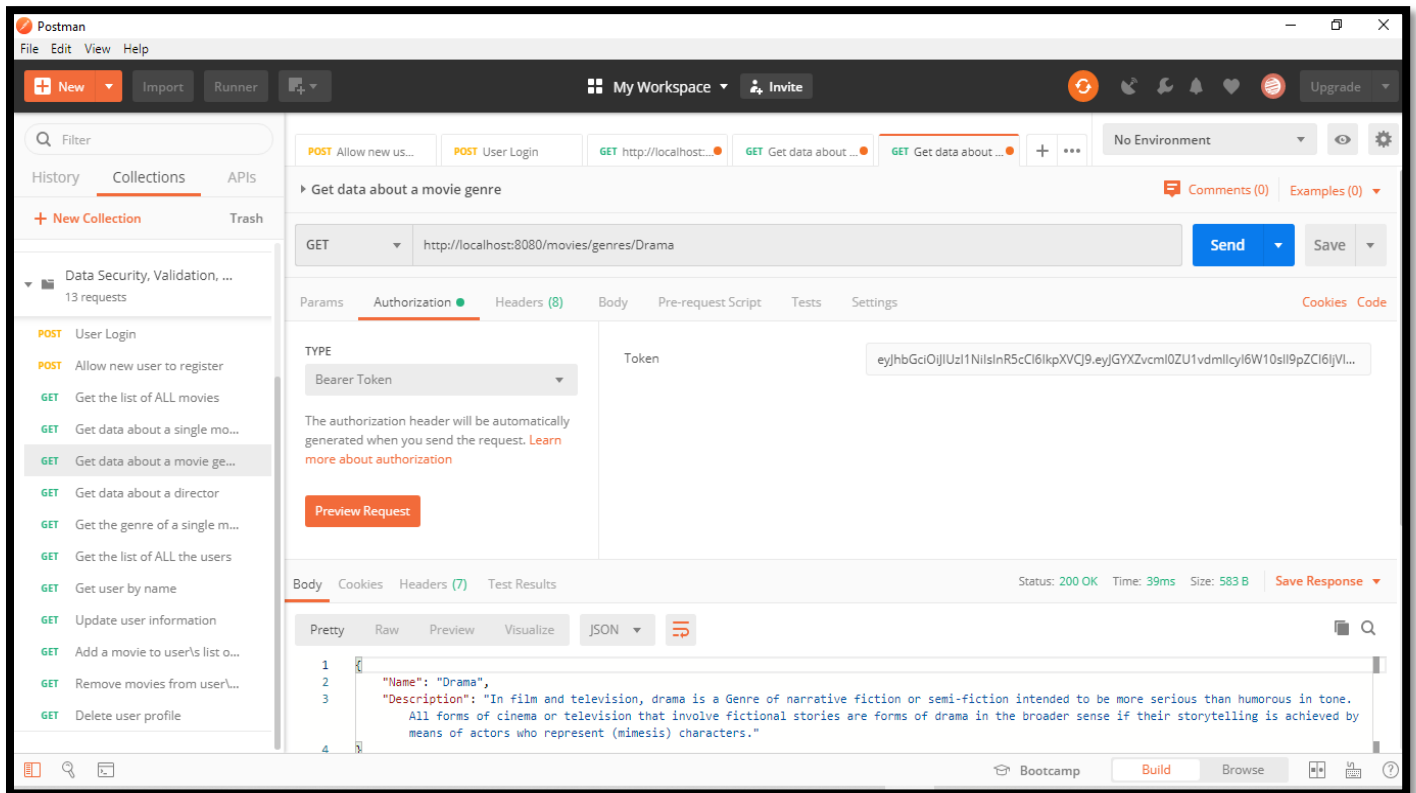
Two: Return data (description, genre, director, image URL, whether it's featured or not) about a single movie by title to the user:



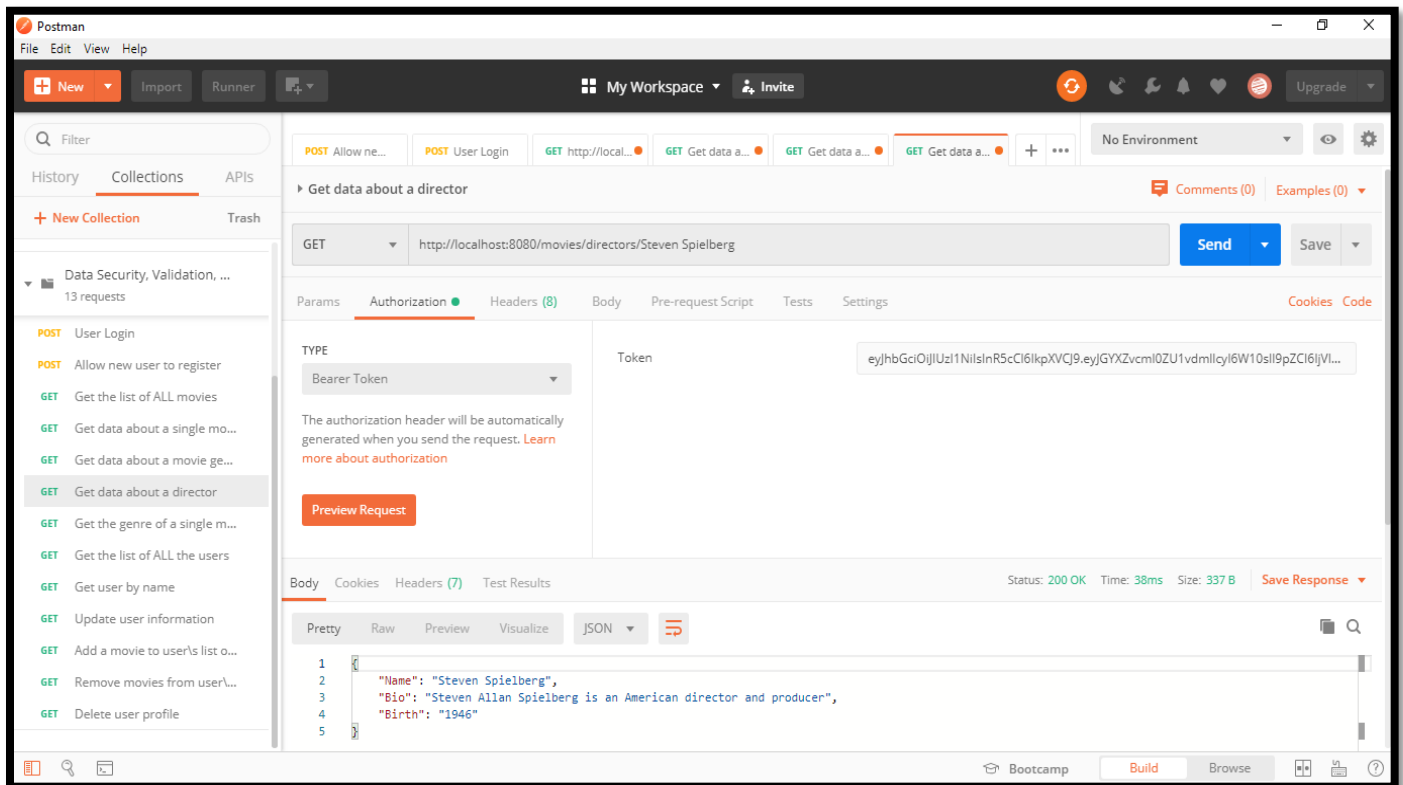
Three: Return data about a genre by the movie title (e.g., “Gladiator”):



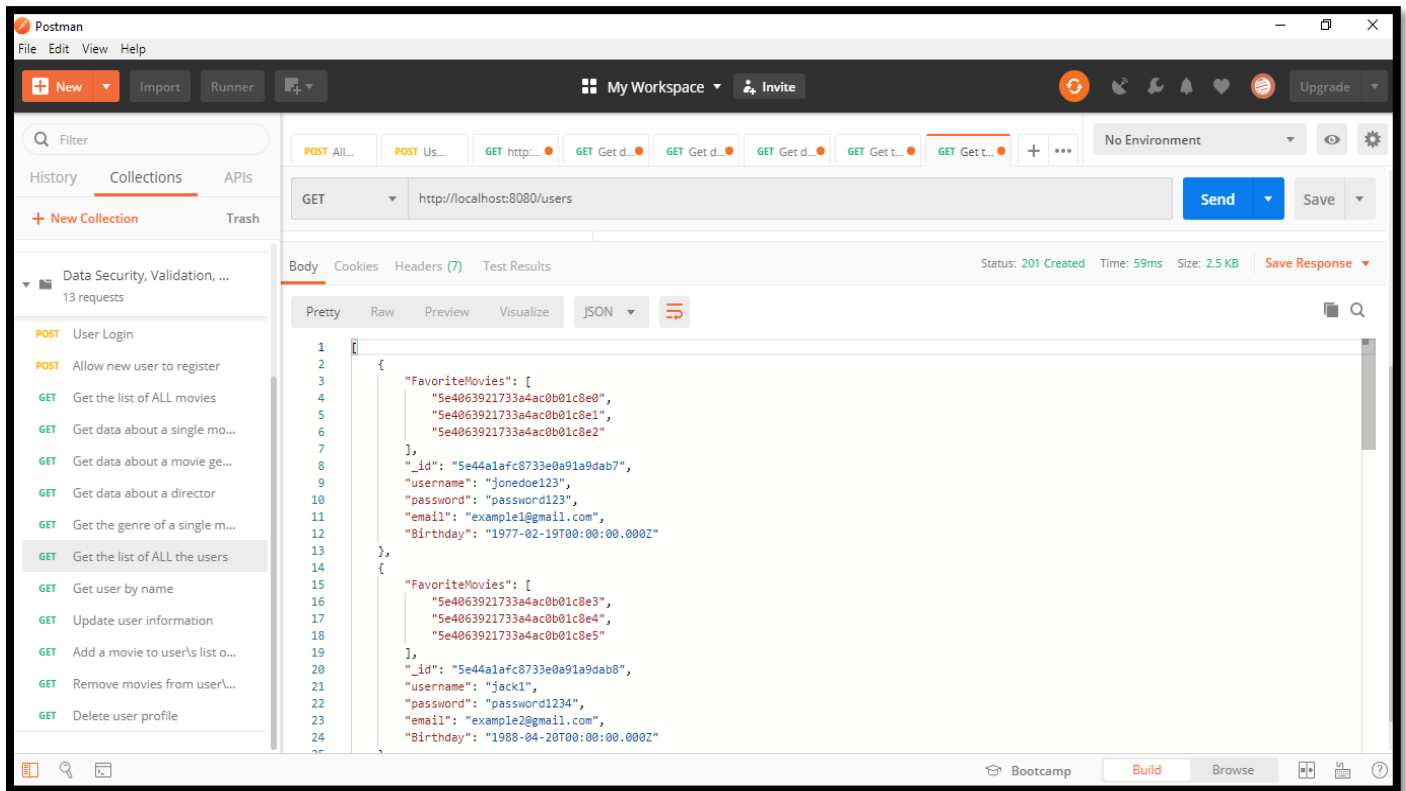
Four: Return data about a genre (description) by the genre Name (e.g., “Drama”):



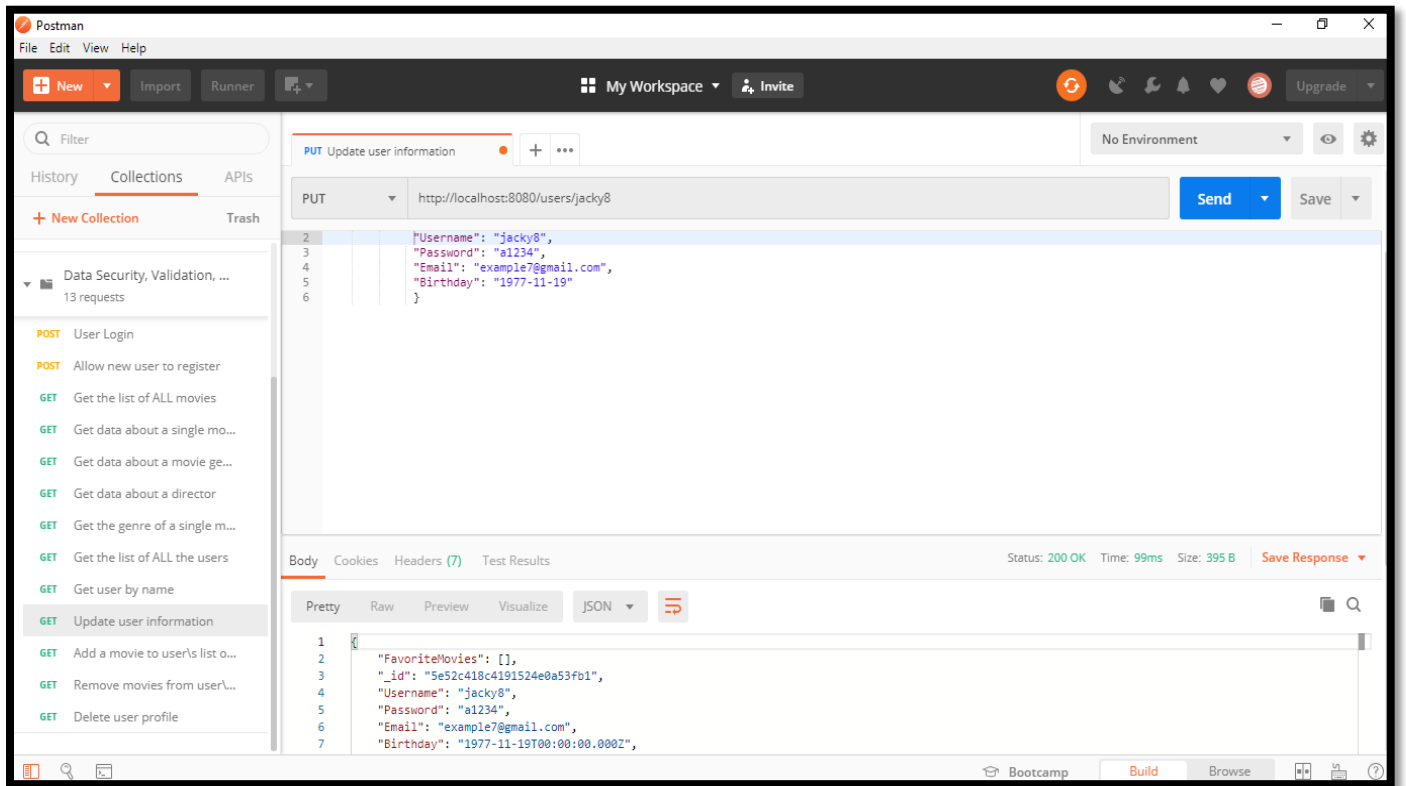
Five: Return data about a director (bio, birth year, death year) by name:



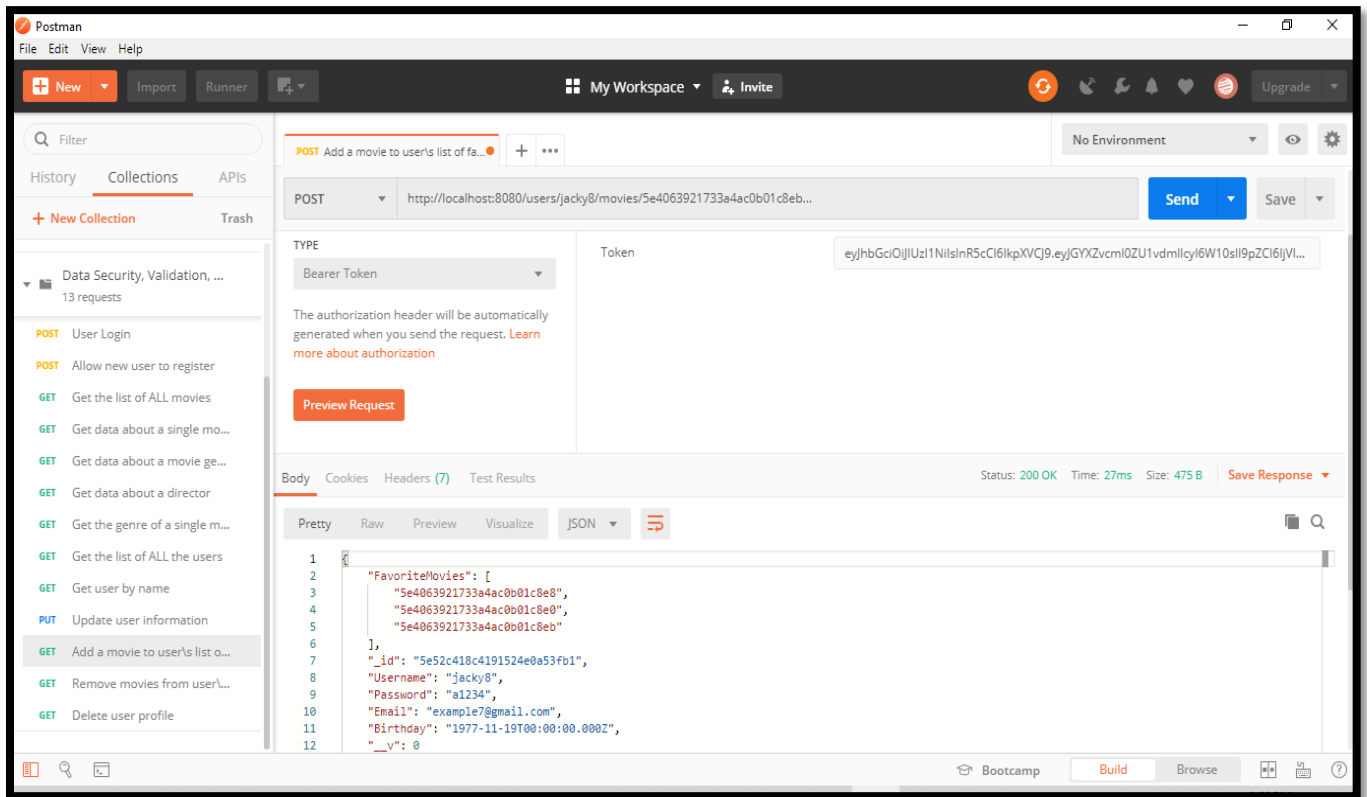
Six: Return a list of ALL users:



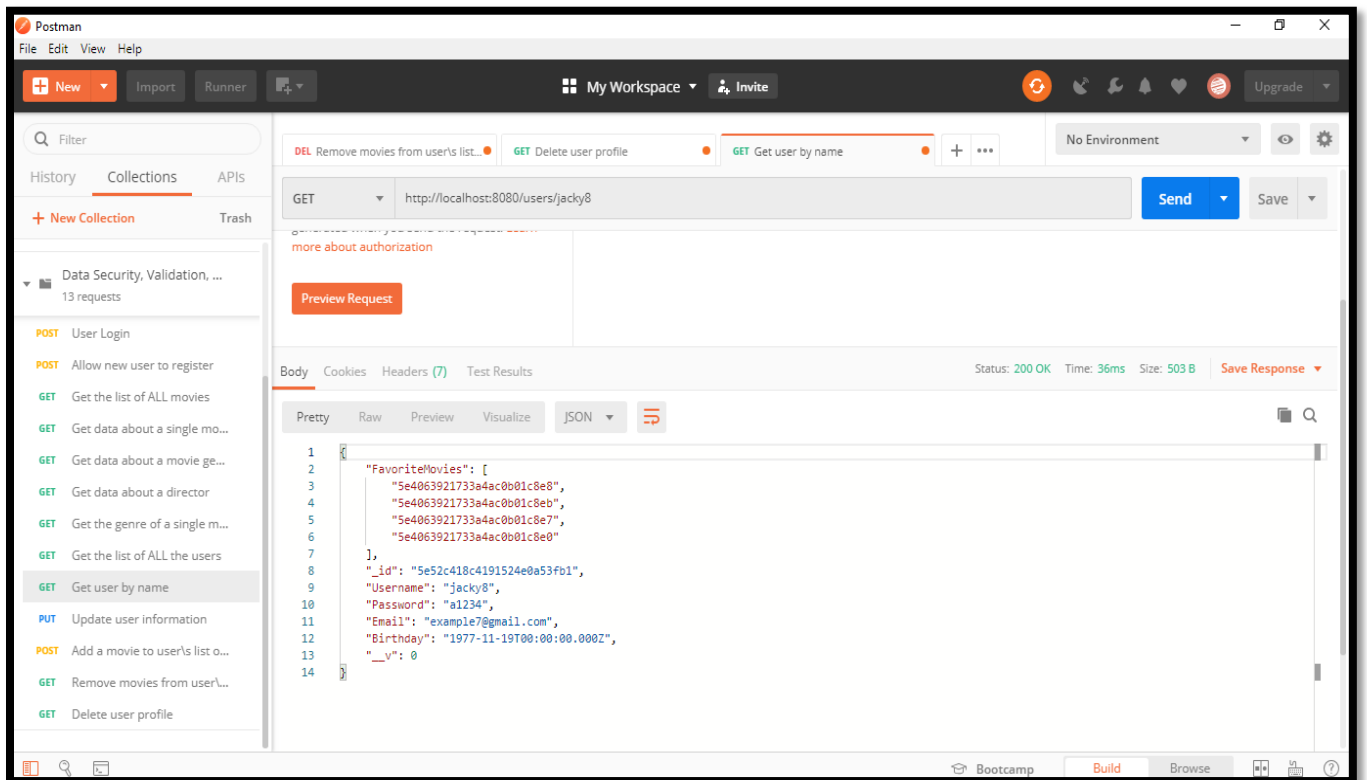
Seven: Allow users to update their user info (username, password, email, date of birth):



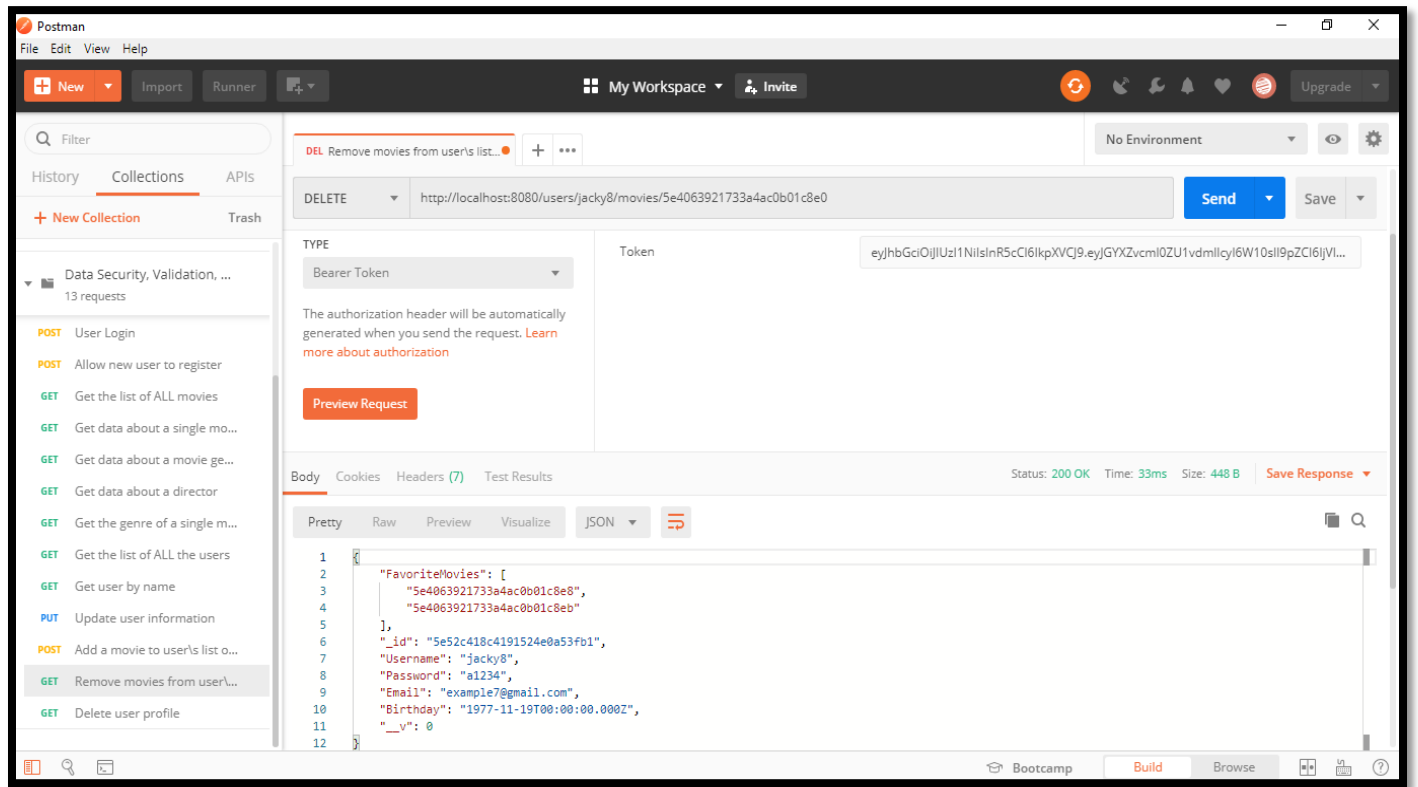
Eight: Allow users to add a movie to their list of favorites:



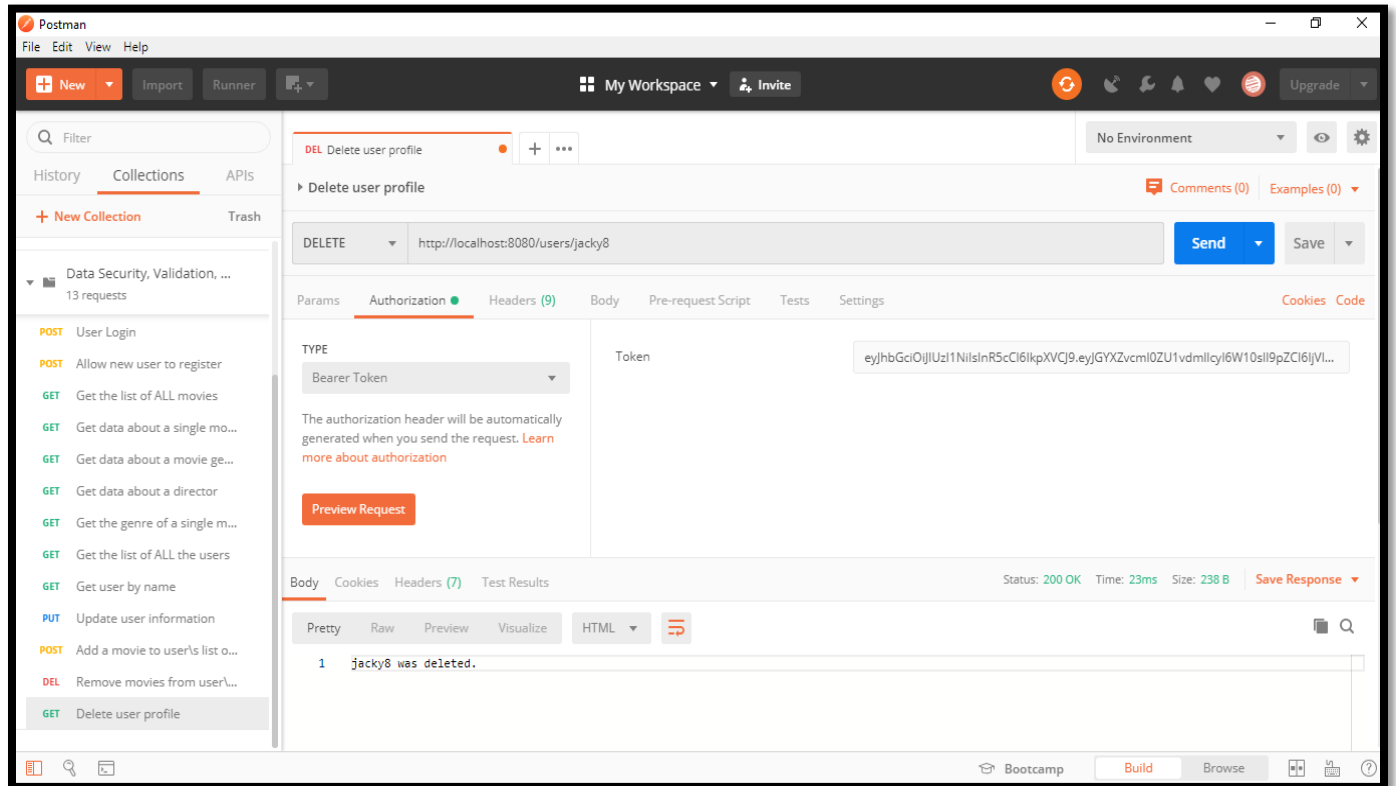
Nine: Get user by Name:



Ten: Allow users to remove a movie from their list of favorites:



Eleven: Allow existing users to deregister with authorization token:



Exercises 2.10b

HEROKU

Installed Heroku via command prompt:

❏ Select Command Prompt - heroku login -i

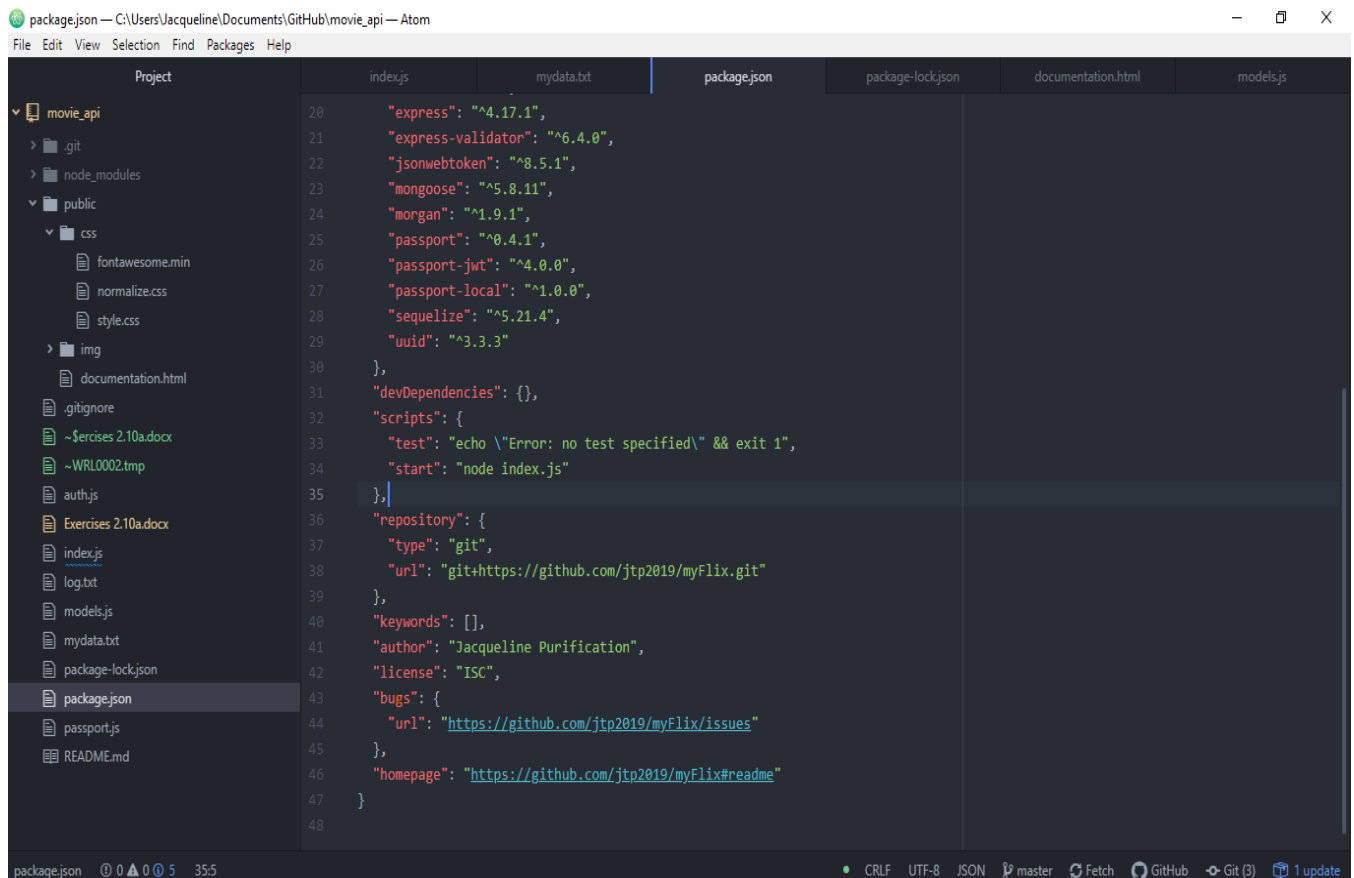
```
Microsoft Windows [Version 10.0.18362.175]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Jacqueline>heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/a98993cf-9fab-409d-b136-2c18c68258dc
Logging in... done
Logged in as jackpr77@gmail.com

C:\Users\Jacqueline>heroku --version
heroku/7.38.2 win32-x86 node-v12.13.0

C:\Users\Jacqueline>heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/3a2f0d95-a70d-424f-b9f4-0b3847b57e68
Logging in... done
Logged in as jackpr77@gmail.com
```

Updated package.json added "start": "node index.js" :

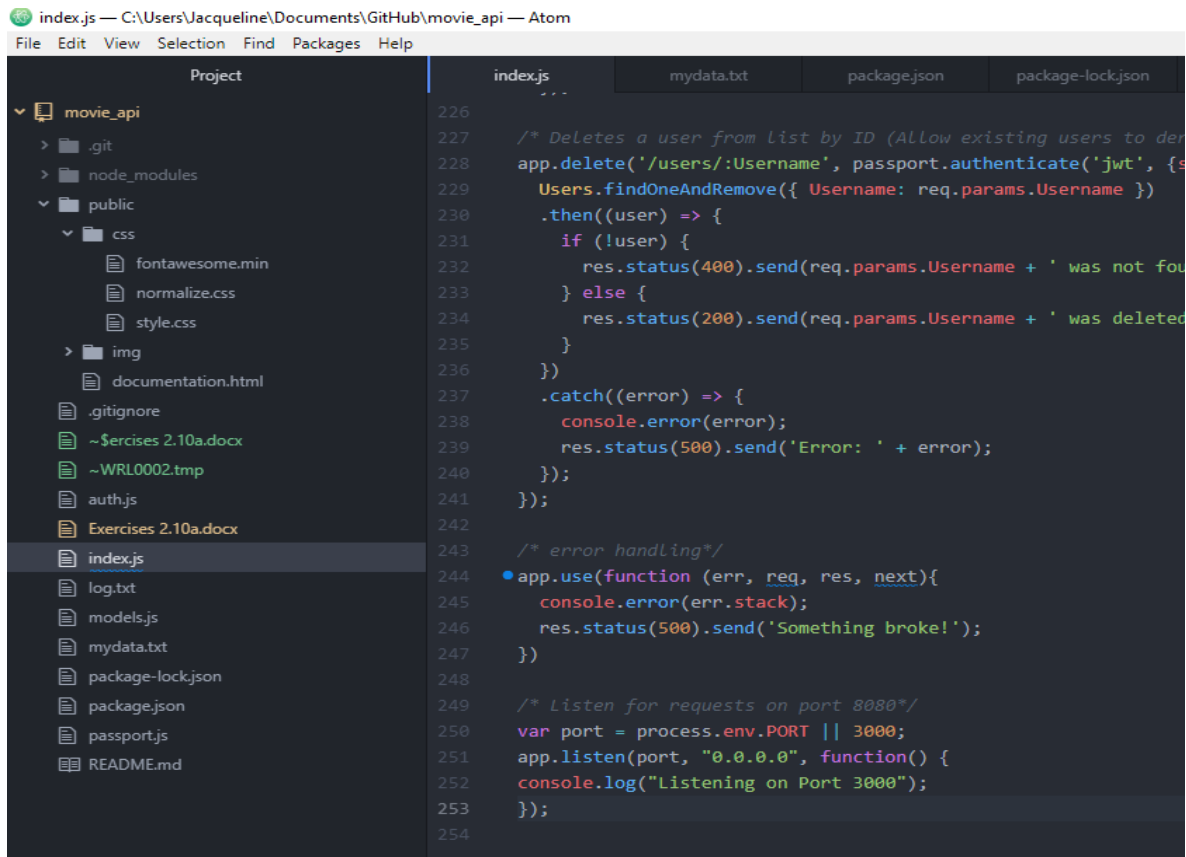


The screenshot shows a code editor with the 'package.json' file open. The file is located at 'C:\Users\Jacqueline\Documents\GitHub\movie_api'. The editor has a dark theme and a sidebar on the left showing the project structure. The 'package.json' file contains the following JSON:

```
{
  "name": "movie_api",
  "version": "1.0.0",
  "description": "A simple movie API",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node index.js"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/jtp2019/myFlix.git"
  },
  "keywords": [],
  "author": "Jacqueline Purification",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/jtp2019/myFlix/issues"
  },
  "homepage": "https://github.com/jtp2019/myFlix#readme"
}
```

The sidebar on the left shows the project structure with folders like '.git', 'node_modules', 'public', 'css', 'img', and files like 'documentation.html', '.gitignore', 'Exercises 2.10a.docx', 'index.js', 'log.txt', 'models.js', 'mydata.txt', 'package-lock.json', 'passport.js', and 'README.md'. The 'package.json' file is currently selected and its content is displayed in the main editor area.

Updated app.listen();

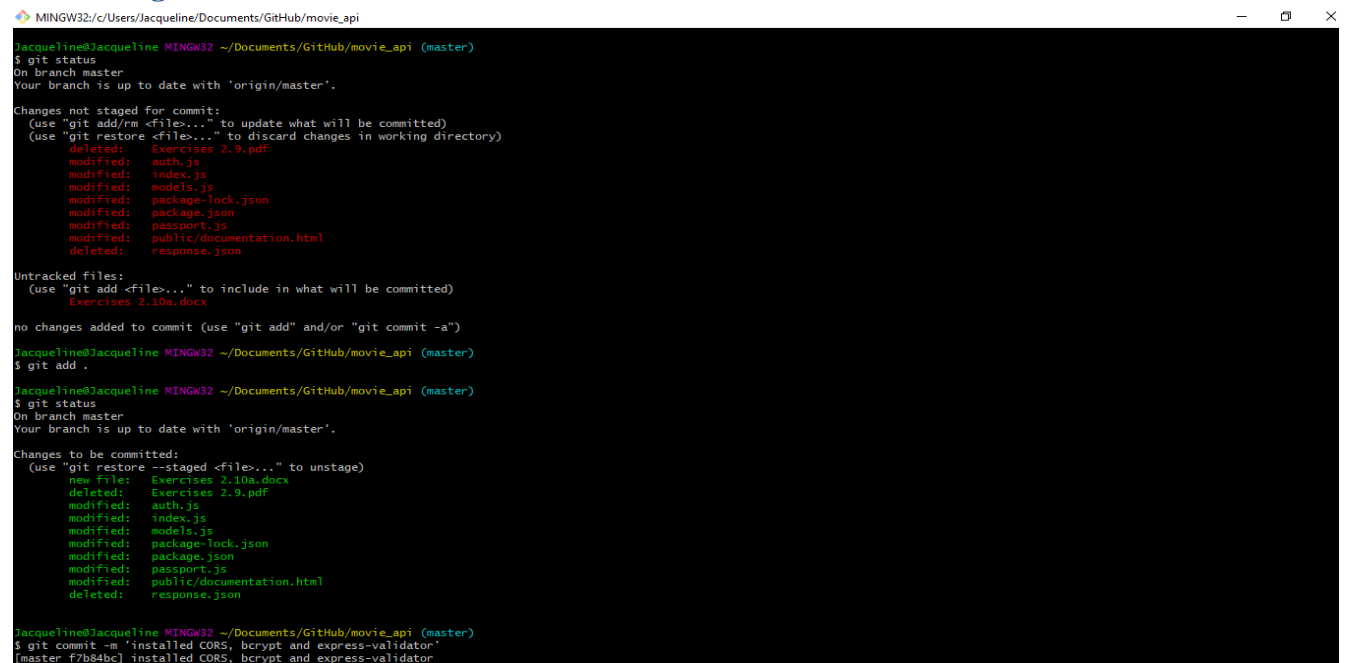


```
index.js — C:\Users\Jacqueline\Documents\GitHub\movie_api — Atom
File Edit View Selection Find Packages Help

Project
  movie_api
    .git
    node_modules
    public
      css
        fontawesome.min
        normalize.css
        style.css
      img
        documentation.html
    .gitignore
    ~$ercises 2.10a.docx
    ~WRL0002.tmp
    auth.js
    Exercises 2.10a.docx
    index.js
    log.txt
    models.js
    mydata.txt
    package-lock.json
    package.json
    passport.js
    README.md

index.js
226
227  /* Deletes a user from list by ID (Allow existing users to der
228  app.delete('/users/:Username', passport.authenticate('jwt', {s
229    Users.findOneAndRemove({ Username: req.params.Username })
230    .then((user) => {
231      if (!user) {
232        res.status(400).send(req.params.Username + ' was not fou
233      } else {
234        res.status(200).send(req.params.Username + ' was deleted
235      }
236    })
237    .catch((error) => {
238      console.error(error);
239      res.status(500).send('Error: ' + error);
240    });
241  });
242
243  /* error handling*/
244  app.use(function (err, req, res, next){
245    console.error(err.stack);
246    res.status(500).send('Something broke!');
247  })
248
249  /* Listen for requests on port 3000*/
250  var port = process.env.PORT || 3000;
251  app.listen(port, "0.0.0.0", function() {
252    console.log("Listening on Port 3000");
253  });
254
```

Push changes to Git and heroku master:



```
MINGW32/c/Users/Jacqueline/Documents/GitHub/movie_api
Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    Exercises 2.9.pdf
    modified:   auth.js
    modified:   index.js
    modified:   models.js
    modified:   package-lock.json
    modified:   package.json
    modified:   passport.js
    modified:   public/documentation.html
    deleted:    response.json

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Exercises 2.10a.docx

no changes added to commit (use "git add" and/or "git commit -a")
Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ git add .
Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Exercises 2.10a.docx
    deleted:    Exercises 2.9.pdf
    modified:   auth.js
    modified:   index.js
    modified:   models.js
    modified:   package-lock.json
    modified:   package.json
    modified:   passport.js
    modified:   public/documentation.html
    deleted:    response.json

Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ git commit -m 'installed CORS, bcrypt and express-validator'
[master f7b84bc] installed CORS, bcrypt and express-validator
```

Create and deploy my app on Heroku:

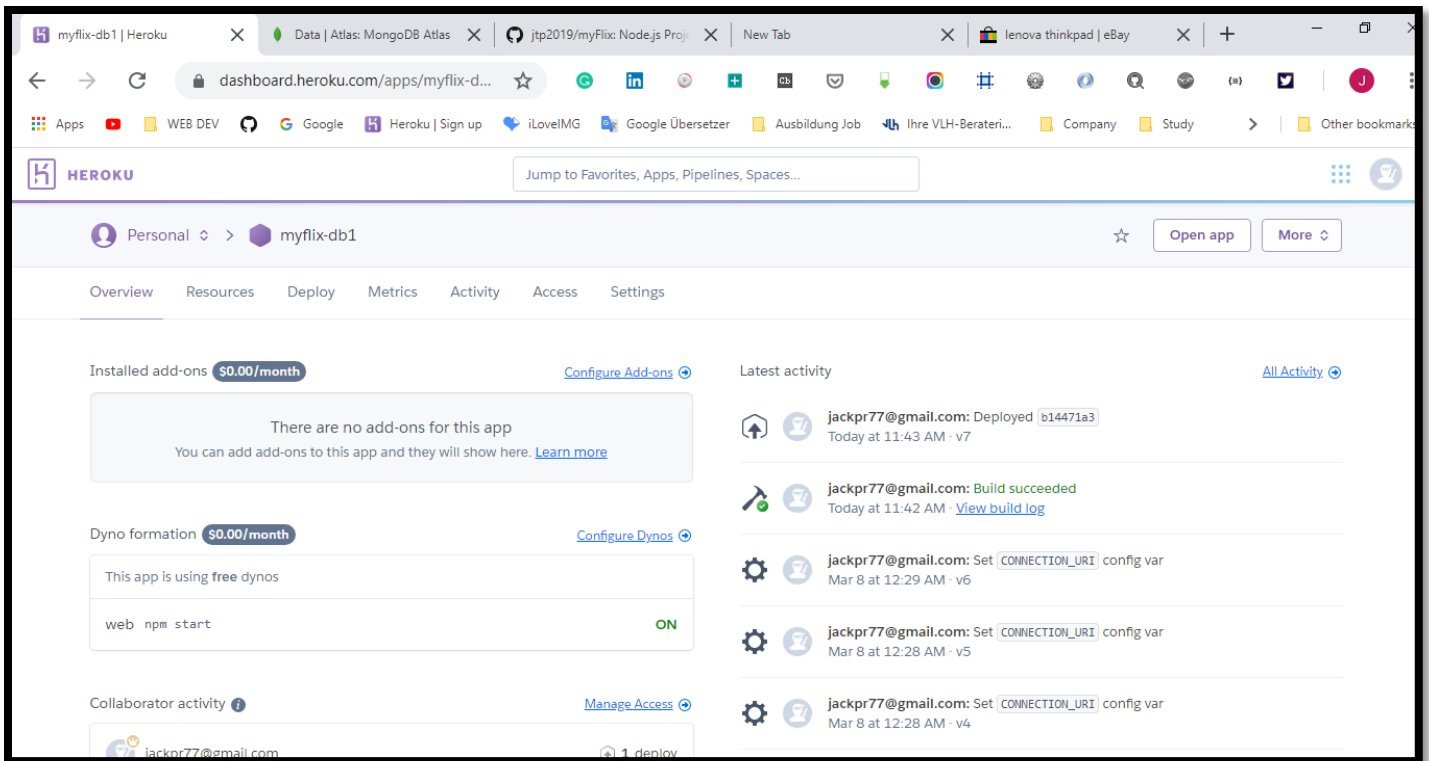
MINGW32:/c/Users/Jacqueline/Documents/GitHub/movie_api

```
Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ git push heroku
Enumerating objects: 109, done.
Counting objects: 100% (109/109), done.
Delta compression using up to 4 threads
Compressing objects: 100% (105/105), done.
Writing objects: 100% (109/109), 8.93 MiB | 91.00 KiB/s, done.
Total 109 (delta 39), reused 0 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Node.js app detected
remote:
remote: -----> Creating runtime environment
remote:
remote:       NPM_CONFIG_LOGLEVEL=error
remote:       NODE_ENV=production
remote:       NODE_MODULES_CACHE=true
remote:       NODE_VERBOSE=false
remote:
remote: -----> Installing binaries
remote:       engines.node (package.json): unspecified
remote:       engines.npm (package.json):  unspecified (use default)
remote:
remote:       Resolving node version 12.x...
remote:       Downloading and installing node 12.16.1...
remote:       Using default npm version: 6.13.4
remote:
remote: -----> Installing dependencies
remote:       Installing node modules (package.json + package-lock)
remote:
remote:       > bcrypt@4.0.0 install /tmp/build_52972d4e85379955ede1926c4b87e57a/node_modules/bcrypt
remote:       > node-pre-gyp install --fallback-to-build
remote:
remote:       [bcrypt] Success: "/tmp/build_52972d4e85379955ede1926c4b87e57a/node_modules/bcrypt/lib/binding/napi-v3/bcrypt_lib.node" is installed via remote
remote:       added 196 packages from 341 contributors and audited 528 packages in 6.086s
remote:
remote:       2 packages are looking for funding
remote:       run 'npm fund' for details
remote:
remote:       found 0 vulnerabilities
remote:
remote: -----> Build
remote:
remote: -----> Caching build
remote:       - node_modules
remote:
remote: -----> Pruning devDependencies
```

MINGW32:/c/Users/Jacqueline/Documents/GitHub/movie_api

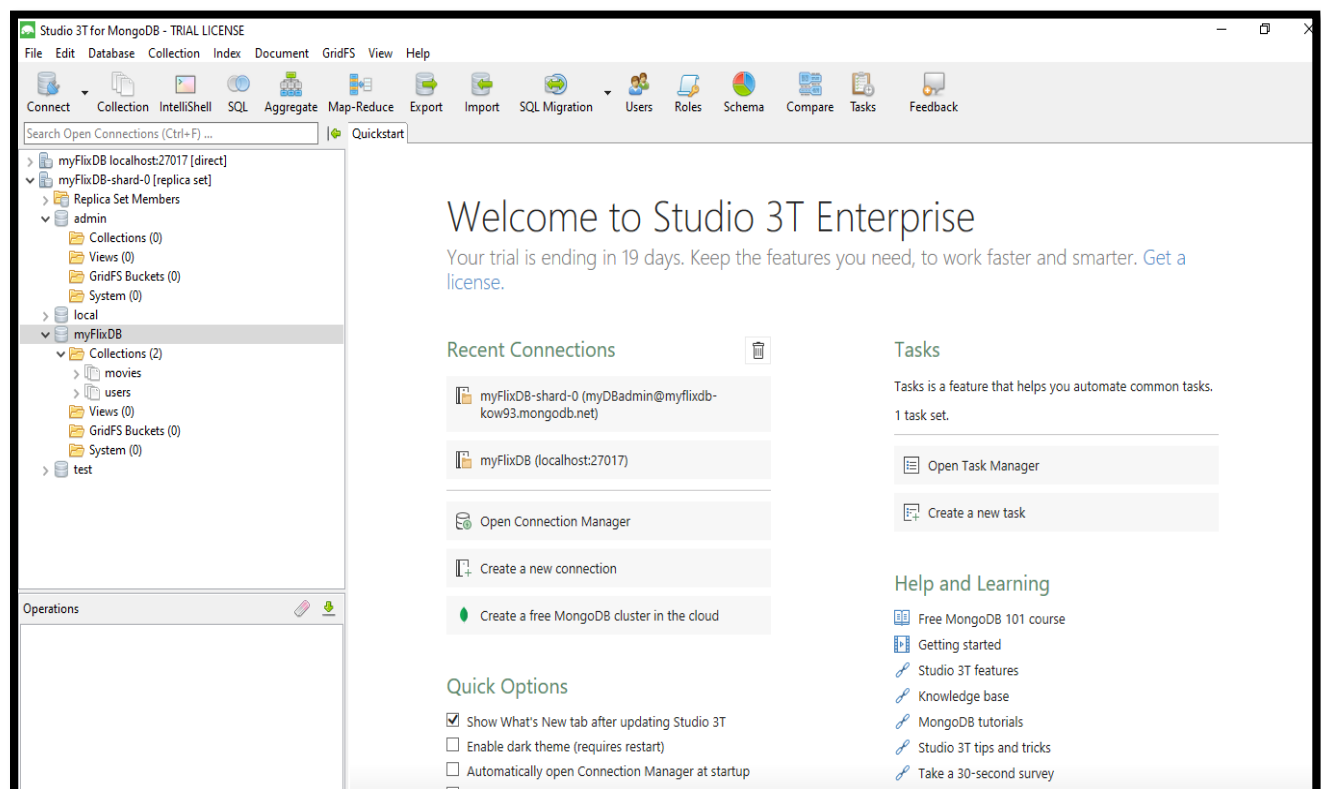
```
remote:       Resolving node version 12.x...
remote:       Downloading and installing node 12.16.1...
remote:       Using default npm version: 6.13.4
remote:
remote: -----> Installing dependencies
remote:       Installing node modules (package.json + package-lock)
remote:
remote:       > bcrypt@4.0.0 install /tmp/build_52972d4e85379955ede1926c4b87e57a/node_modules/bcrypt
remote:       > node-pre-gyp install --fallback-to-build
remote:
remote:       [bcrypt] Success: "/tmp/build_52972d4e85379955ede1926c4b87e57a/node_modules/bcrypt/lib/binding/napi-v3/bcrypt_lib.node" is installed via remote
remote:       added 196 packages from 341 contributors and audited 528 packages in 6.086s
remote:
remote:       2 packages are looking for funding
remote:       run 'npm fund' for details
remote:
remote:       found 0 vulnerabilities
remote:
remote: -----> Build
remote:
remote: -----> Caching build
remote:       - node_modules
remote:
remote: -----> Pruning devDependencies
remote:       audited 528 packages in 1.73s
remote:
remote:       2 packages are looking for funding
remote:       run 'npm fund' for details
remote:
remote:       found 0 vulnerabilities
remote:
remote: -----> Build succeeded!
remote: -----> Discovering process types
remote:       Procfile declares types   -> (none)
remote:       Default types for buildpack -> web
remote:
remote: -----> Compressing...
remote:       Done: 28.7M
remote: -----> Launching...
remote:       Released v7
remote:       https://myflix-dbl.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/myflix-dbl.git
 * [new branch]    master -> master

Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
```



Upload database to MongoDB Atlas via Studio 3T

1st Step: Installed Studio 3T for 32 bit Win 10, Connect S3T with mongodb database (device) and added all collections to **myFlixDB (localhost 27027)** folder. After successfully getting the collections export & execute those data in Json format to my PC. Which will automatic created a folder in my PC named myFlixDB.



2nd Step: Open account on MongoDB atlas, created a cluster named myFlixBDB, and setup database & network access

The screenshot shows the MongoDB Atlas 'Clusters' page for a cluster named 'myFlixBDB'. The left sidebar contains navigation links for ATLAS (Clusters, Data Lake BETA), SECURITY (Database Access, Network Access, Advanced), PROJECT (Access Management, Activity Feed, Alerts, Integrations, Settings), and a 'CONTEXT' section for 'Jacqueline's Org ...'. The main content area displays cluster details: 'Sandbox' tier, 'myFlixBDB' version 4.2.3, 'M0 Sandbox (General)' cluster tier, 'AWS / Frankfurt (eu-central-1)' region, 'Replica Set - 3 nodes' type, and 'None Linked' for the linked stitch app. Performance metrics include 'Operations R: 0 W: 0' at 0.01/s, 'Logical Size 33.2 KB', and 'Connections 13'. A 'Build a New Cluster' button is in the top right, and an 'Upgrade' button is in the bottom right.

The screenshot shows the 'Database Access' page in MongoDB Atlas. The left sidebar is similar to the previous screenshot, with 'Database Access' highlighted under the 'SECURITY' section. The main content area has tabs for 'Database Users' and 'Custom Roles'. A table lists database users with columns for 'User Name', 'Authentication Method', 'MongoDB Roles', and 'Actions'. One user, 'myDBAdmin', is listed with 'SCRAM' authentication and 'atlasAdmin@admin' role. An 'ADD NEW DATABASE USER' button is in the top right.

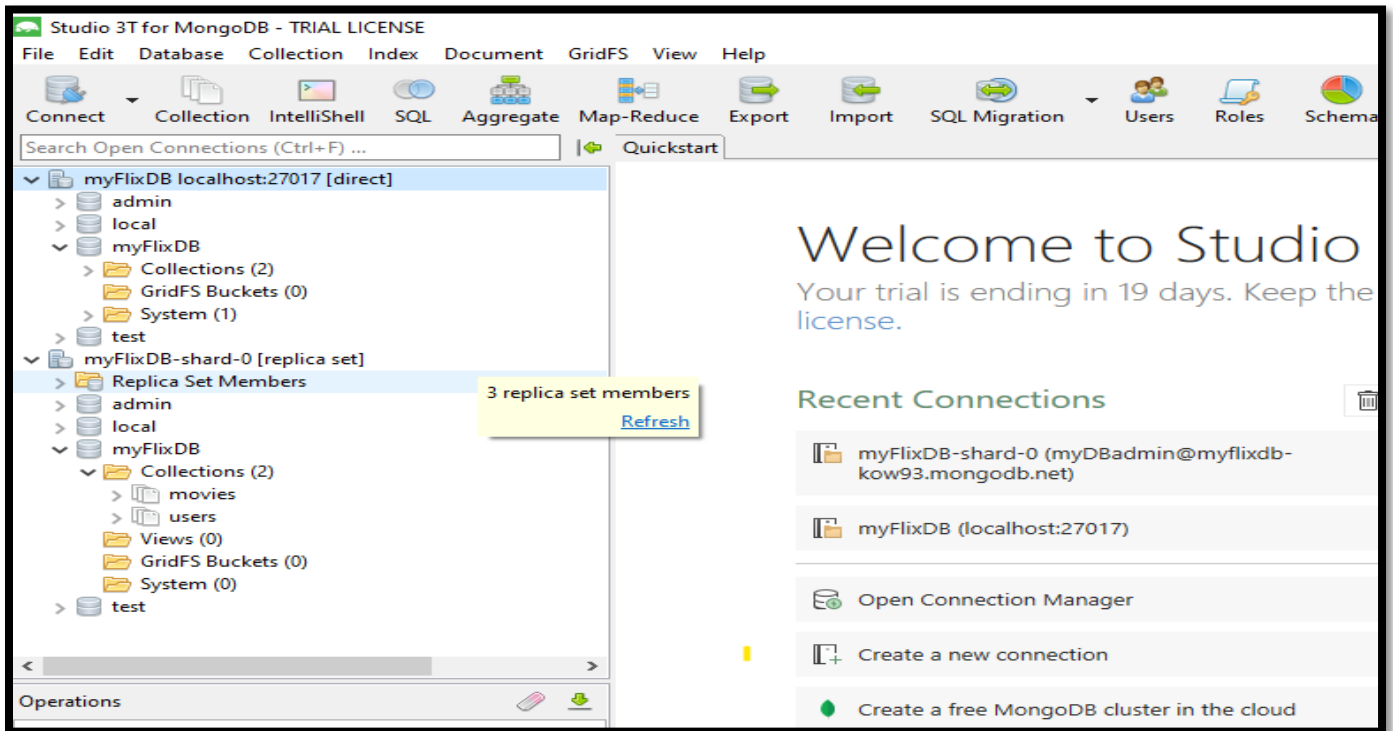
User Name	Authentication Method	MongoDB Roles	Actions
myDBAdmin	SCRAM	atlasAdmin@admin	[EDIT] [DELETE]

The screenshot shows the 'Network Access' page in MongoDB Atlas. The left sidebar is similar, with 'Network Access' highlighted under the 'SECURITY' section. The main content area has tabs for 'IP Whitelist', 'Peering', and 'Private Endpoint'. A message states: 'You will only be able to connect to your cluster from the following list of IP Addresses:'. Below this is a table with columns for 'IP Address', 'Comment', 'Status', and 'Actions'. One IP address, '0.0.0.0/0 (includes your current IP address)', is listed with an 'Active' status. An 'ADD IP ADDRESS' button is in the top right.

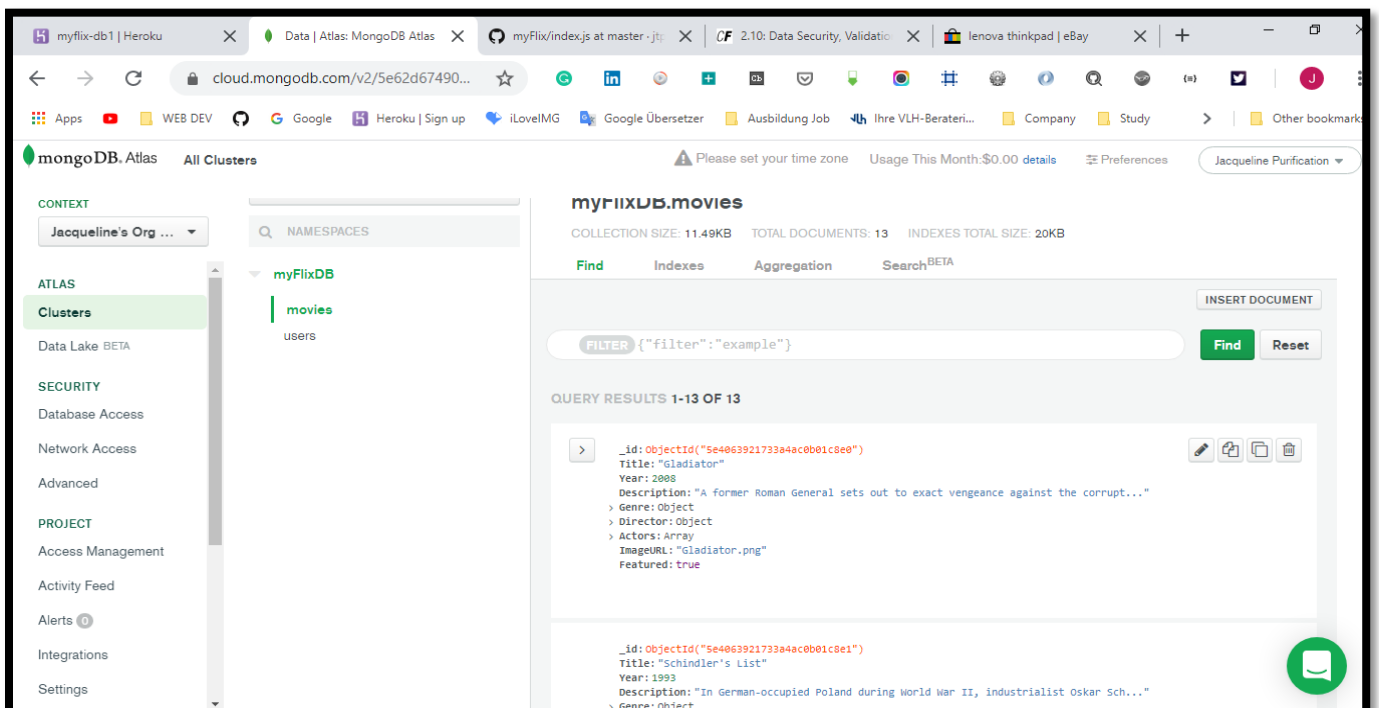
IP Address	Comment	Status	Actions
0.0.0.0/0 (includes your current IP address)		Active	[EDIT] [DELETE]

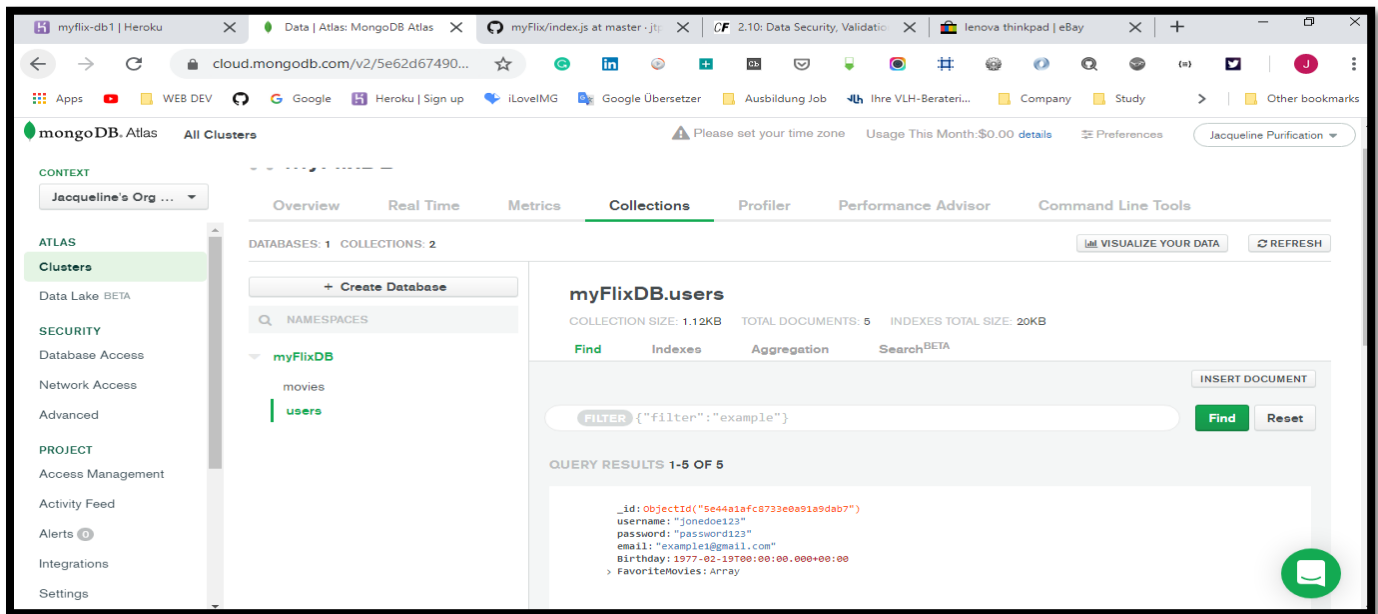
3rd Step: Connected Studio 3T with MonnnoDB Cluster

Open a new connection named **myFlixDB-shard-0** and export & executed all data in Json format from **myFlixDB (localhost 27027)** to my PC in different folder named **myFlixDB** and from that folder import data to the **myFlixDB-shard-0** folder.



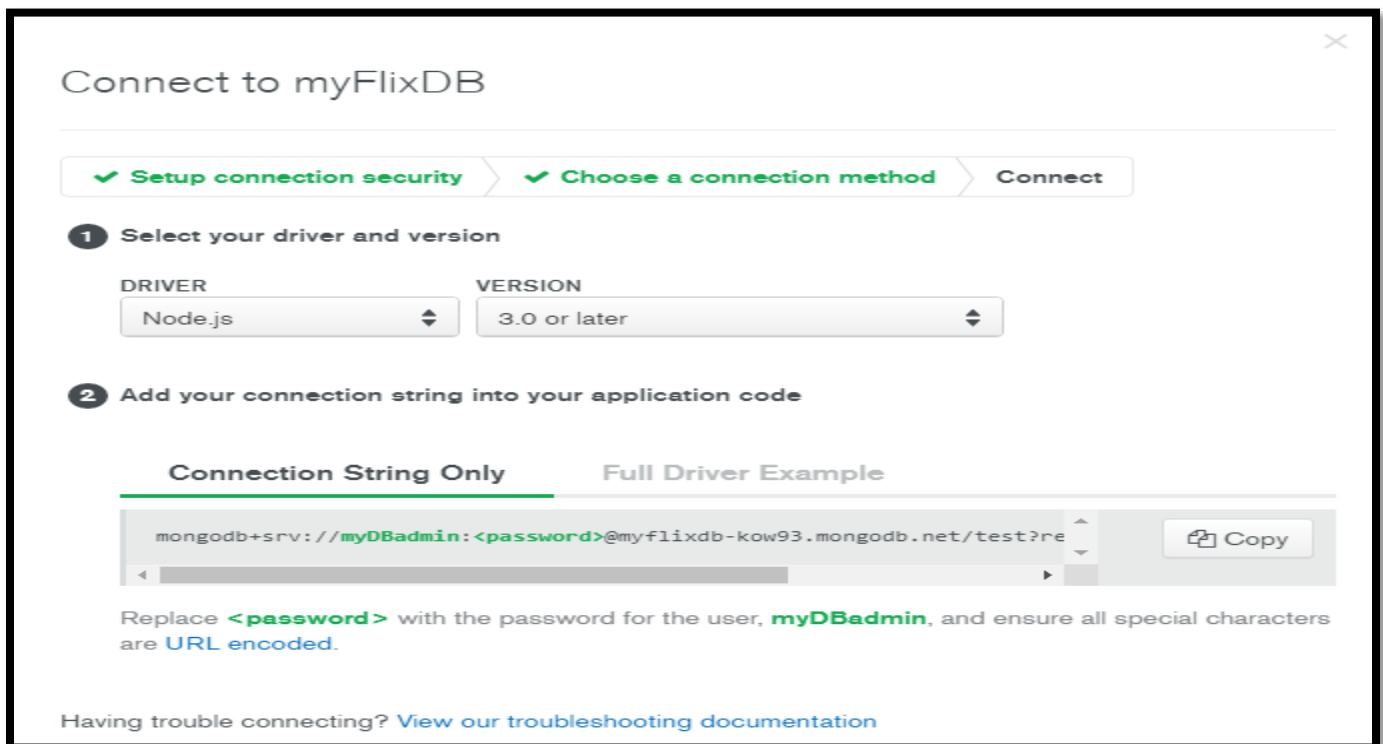
Now the collections are showing on the MongoDB Atlas





Connected MongoDB atlas with Heroku (GitHub)

1st Step: From MongoDB atlas collected connection string
 'mongodb+srv://myDBAdmin:12345@myflixdb-kow93.mongodb.net/test?retryWrites=true&w=majority'
 changed the format to `mongoose.connect(process.env.'mongodb+srv://myDBAdmin:12345@myflixdb-kow93.mongodb.net/test?retryWrites=true&w=majority', { useNewUrlParser: true, useUnifiedTopology: true });` and added that URL to index.js and paste it into Heroku app Config Vars.



HEROKU
Jump to Favorites, Apps, Pipelines, Spaces...

App Information

App Name
myflix-db1

Region
United States

Stack
heroku-18

Framework
Node.js

Slug size
28.7 MiB of 500 MiB

Heroku git URL
https://git.heroku.com/myflix-db1.git

Config Vars

Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.

Config Vars

Hide Config Vars

KEY
VALUE
Add

CONNECTION_URI
mongoose.connect(process.env.'mongodb://'

2nd Step: Push all changes to Heroku and GitHub master.

```

MINGW32:/c/Users/Jacqueline/Documents/GitHub/movie_api

Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ git add .

Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ git commit -am "Correction port connection"
[master ec1eb00] Correction port connection
1 file changed, 1 insertion(+), 1 deletion(-)

Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ git push heroku master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 309 bytes | 103.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Node.js app detected
remote:
remote: -----> Creating runtime environment
remote:
remote:       NPM_CONFIG_LOGLEVEL=error
remote:       NODE_ENV=production
remote:       NODE_MODULES_CACHE=true
remote:       NODE_VERBOSE=false
remote:
remote: -----> Installing binaries
remote:       engines.node (package.json):  unspecified
remote:       engines.npm (package.json):   unspecified (use default)
remote:
remote:       Resolving node version 12.x...
remote:       Downloading and installing node 12.16.1...
remote:       Using default npm version: 6.13.4
remote:
remote: -----> Restoring cache
remote:       - node_modules
remote:
remote: -----> Installing dependencies
remote:       Installing node modules (package.json + package-lock)
remote:       audited 528 packages in 1.667s
remote:
remote:       2 packages are looking for funding
remote:       run `npm fund` for details
remote:
remote:       found 0 vulnerabilities
remote:

```


3rd Step: Checked Heroku logs

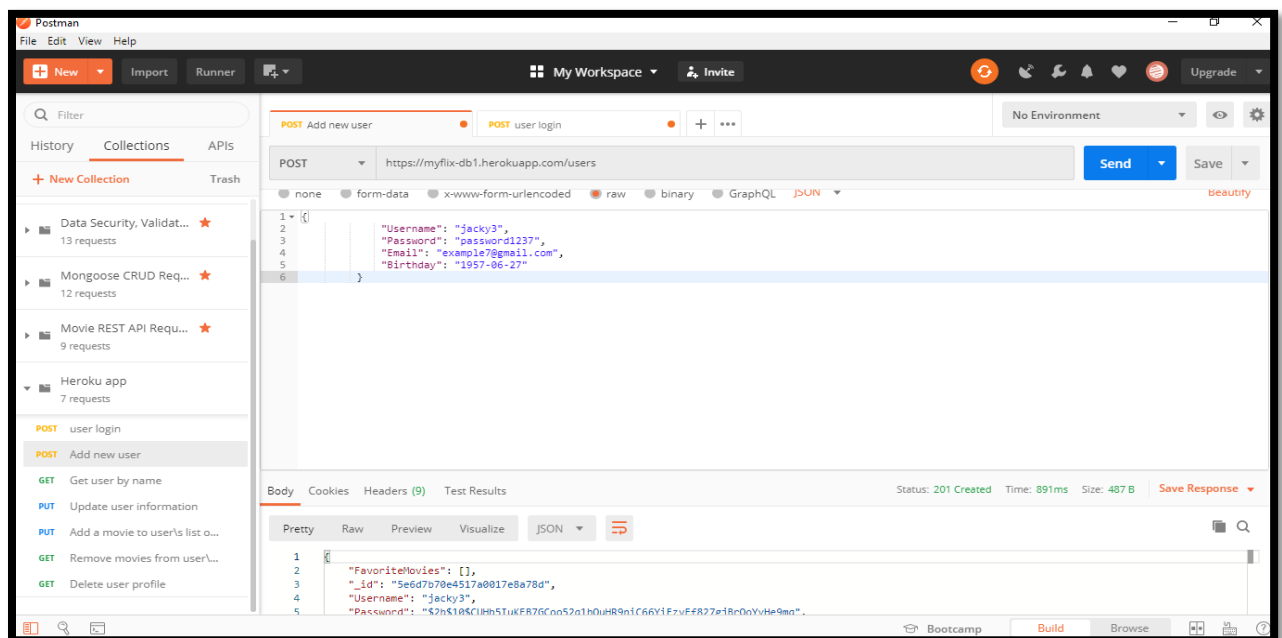
```
MINGW32/c/Users/Jacqueline/Documents/GitHub/movie_api

jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 309 bytes | 103.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
Remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/jtp2019/myFlix.git
   1bc47a2..ec1eb00  master -> master

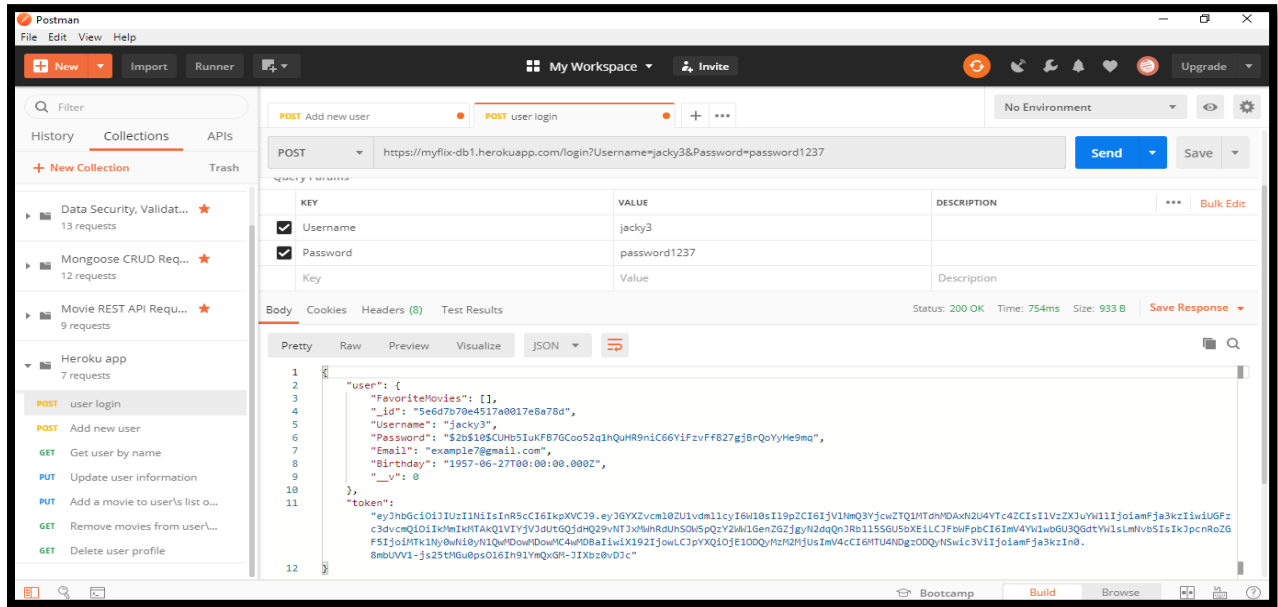
jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ heroku logs --app myFlix-dbl
2020-03-13T19:49:29.863834+00:00 app[web.1]: (node:23) UnhandledPromiseRejectionWarning: MongooseTimeoutError: Server select
2020-03-13T19:49:29.863853+00:00 app[web.1]: at new MongooseTimeoutError (/app/node_modules/mongoose/lib/error/timeout.js:2
2020-03-13T19:49:29.863854+00:00 app[web.1]: at NativeConnection.Connection.openUri (/app/node_modules/mongoose/lib/connect
2020-03-13T19:49:29.863854+00:00 app[web.1]: at Mongoose.connect (/app/node_modules/mongoose/lib/index.js:332:15)
2020-03-13T19:49:29.863855+00:00 app[web.1]: at Object.<anonymous> (/app/index.js:29:10)
2020-03-13T19:49:29.863855+00:00 app[web.1]: at Module._compile (internal/modules/cjs/loader.js:1158:30)
2020-03-13T19:49:29.863855+00:00 app[web.1]: at Object.Module._extensions..js (internal/modules/cjs/loader.js:1178:10)
2020-03-13T19:49:29.863856+00:00 app[web.1]: at Module.load (internal/modules/cjs/loader.js:1002:32)
2020-03-13T19:49:29.863856+00:00 app[web.1]: at Function.Module._load (internal/modules/cjs/loader.js:901:14)
2020-03-13T19:49:29.863856+00:00 app[web.1]: at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:7
2020-03-13T19:49:29.863856+00:00 app[web.1]: at internal/main/run_main_module.js:18:47
2020-03-13T19:49:29.863976+00:00 app[web.1]: (node:23) UnhandledPromiseRejectionWarning: Unhandled promise rejection. This
2020-03-13T19:49:29.863976+00:00 app[web.1]: it a catch block, or by rejecting a promise which was not handled with .catch(). To terminate the node process on unhandled
2020-03-13T19:49:29.863976+00:00 app[web.1]: (see https://nodejs.org/api/cli.html#cli_unhandled_rejections_mode). (rejection id: 2)
2020-03-13T19:56:58.135334+00:00 heroku[router]: at=info method=GET path="/" host=myflix-dbl.herokuapp.com request_id=f0849
2020-03-13T19:56:58.132836+00:00 app[web.1]: 10.63.174.10 - - [13/Mar/2020:19:56:58 +0000] "GET / HTTP/1.1" 404 139
2020-03-13T20:31:49.644843+00:00 heroku[web.1]: Idling
2020-03-13T20:31:49.648056+00:00 heroku[web.1]: State changed from up to down
2020-03-13T20:31:50.662704+00:00 heroku[web.1]: Stopping all processes with SIGTERM
2020-03-13T20:31:50.732020+00:00 heroku[web.1]: Process exited with status 143
2020-03-13T21:32:57.000000+00:00 app[api]: Build started by user jackpr77@gmail.com
2020-03-13T21:33:14.070442+00:00 app[api]: Deploy 1bc47a2c by user jackpr77@gmail.com
2020-03-13T21:33:14.070442+00:00 app[api]: Release v10 created by user jackpr77@gmail.com
2020-03-13T21:33:14.458476+00:00 heroku[web.1]: State changed from down to starting
2020-03-13T21:33:15.000000+00:00 app[api]: Build succeeded
2020-03-13T21:33:16.860821+00:00 heroku[web.1]: Starting process with command 'npm start'
2020-03-13T21:33:19.313949+00:00 app[web.1]:
2020-03-13T21:33:19.313970+00:00 app[web.1]: > movie_api@1.0.0 start /app
2020-03-13T21:33:19.313971+00:00 app[web.1]: > node index.js
2020-03-13T21:33:19.313971+00:00 app[web.1]:
2020-03-13T21:33:20.063160+00:00 app[web.1]: events.js:288
2020-03-13T21:33:20.063188+00:00 app[web.1]: throw er; // Unhandled 'error' event
2020-03-13T21:33:20.063188+00:00 app[web.1]: ^
2020-03-13T21:33:20.063189+00:00 app[web.1]:
```

Postman screenshots (testing code after deploying via Heroku and MongoDB Atlas)

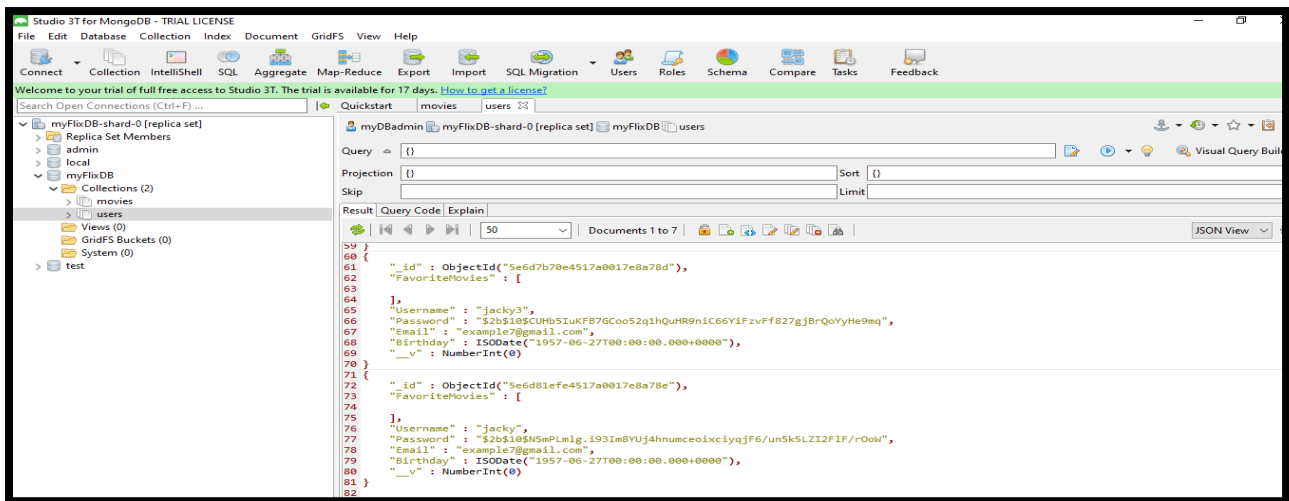
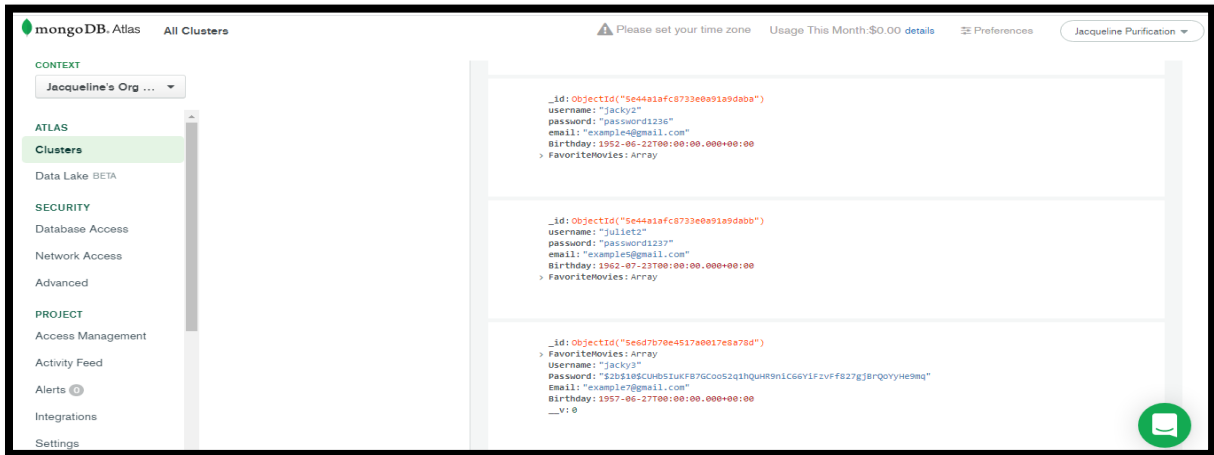
New User Registration



New User login and authorization token

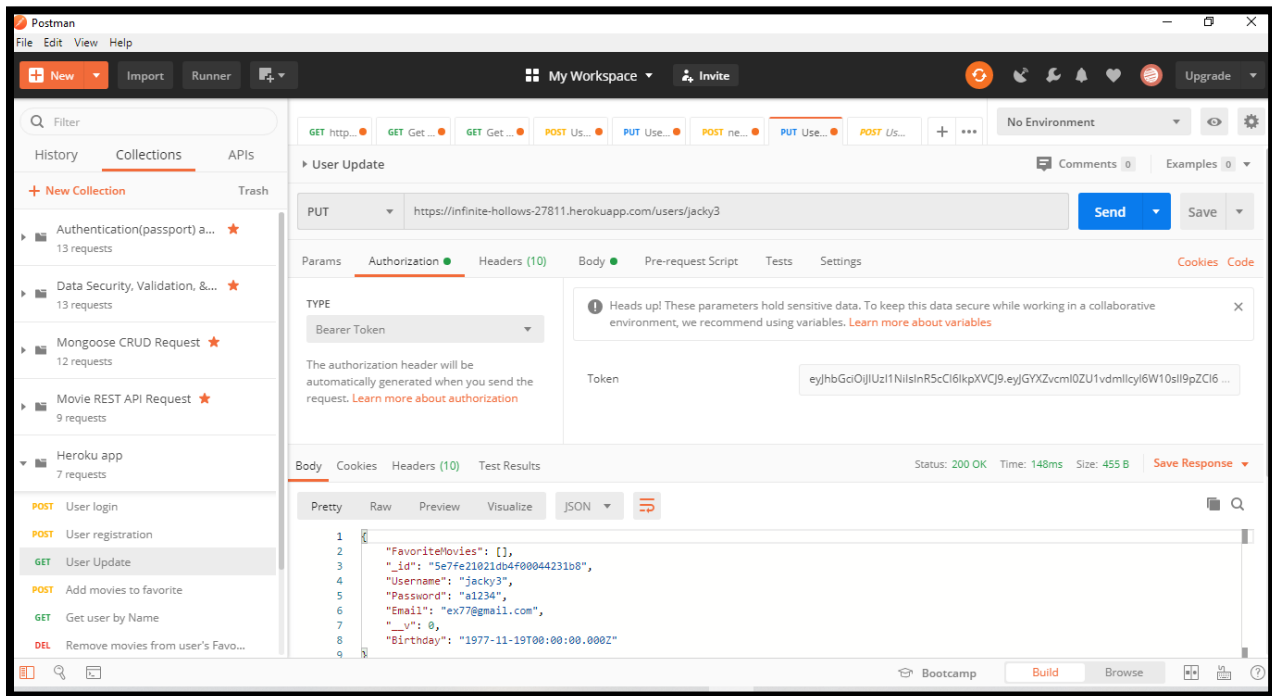


New Users automatically added with MongoDB Atlas and Studio 3T

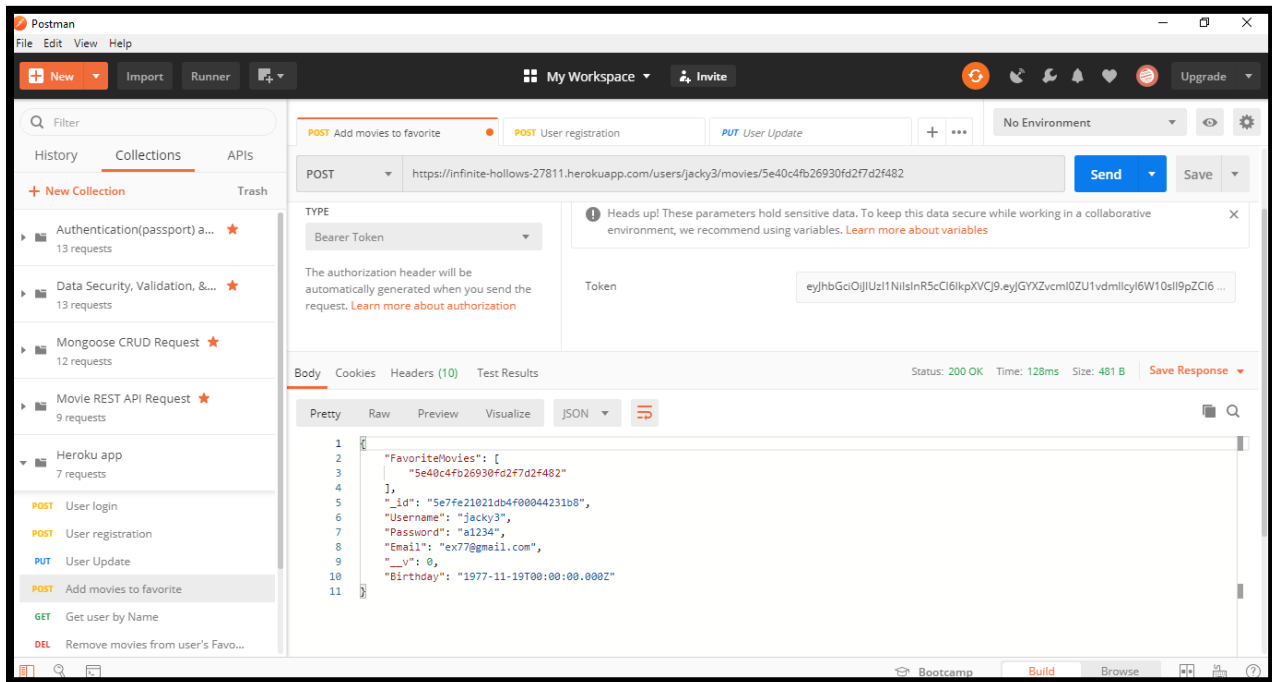


Postman screenshots with authorization token (5 requests)

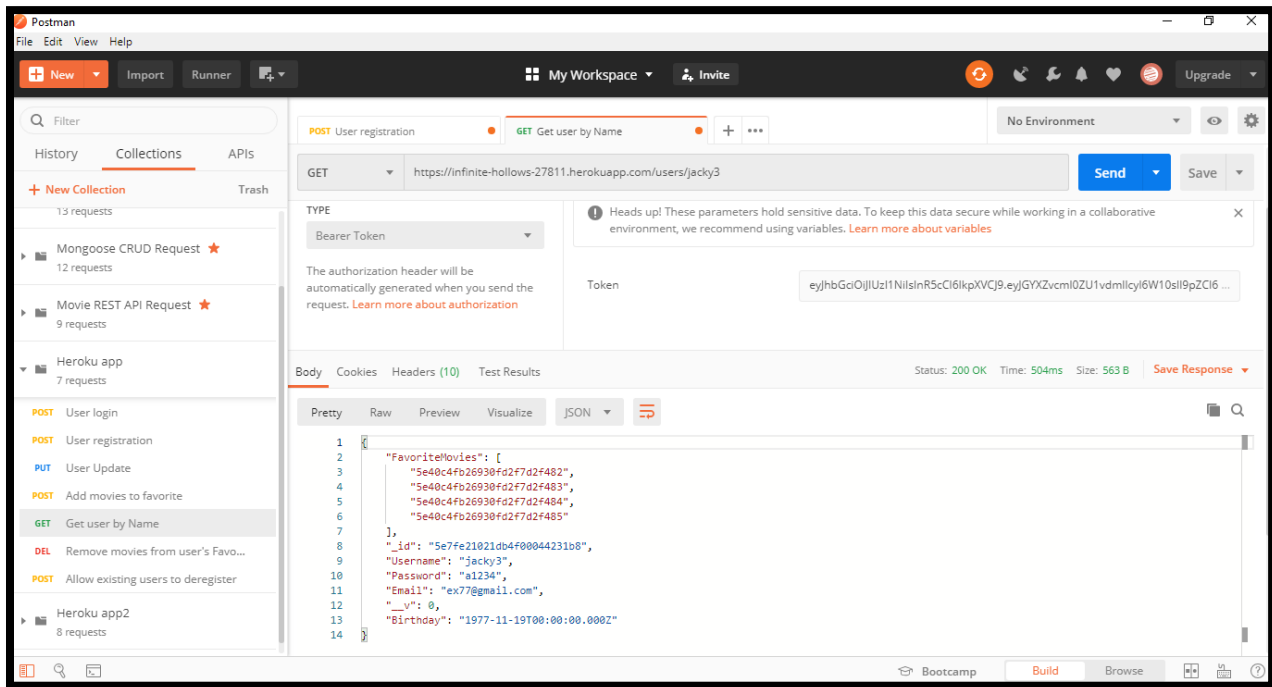
One: Allow users to update their user info (username, password, email, date of birth):



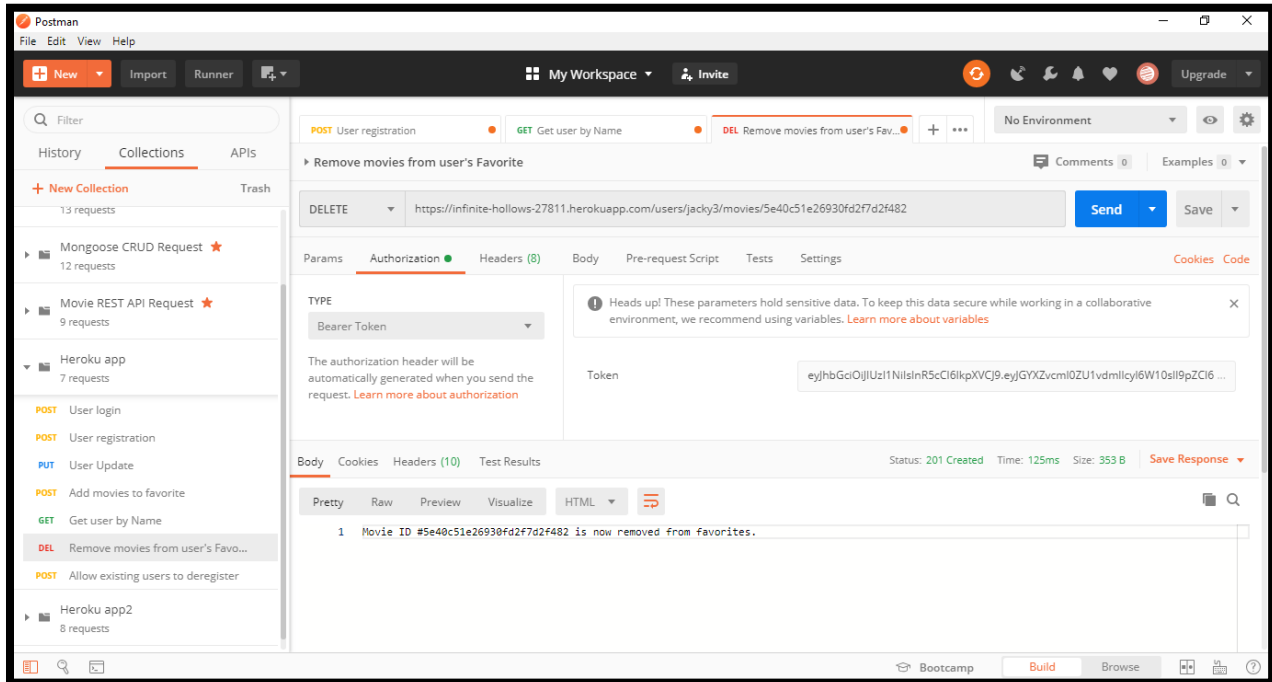
Two: Allow users to add a movie to their list of favorites:



Three: Get user by Name:



Four: Allow users to remove a movie from their list of favorites:



Five: Allow existing users to deregister with authorization token:

