

## Project Setup Instructions

To recreate the solution, follow these steps:

1. Copy the zip submission folder on your machine or clone the project repository from GitHub:  
<https://github.com/sumedhsalvi/optamo-coding-task>.
2. Ensure that you have **Node.js** and **Angular CLI** installed on your machine.
3. Navigate to the optamo-coding-task/movies-search-engine-ui folder and run “**npm install**” to install all the Angular dependencies and packages.
4. Navigate to the optamo-coding-task/movies-search-engine-api folder and run “**pip install -r requirements.txt**” to install all the Flask packages and dependencies.
5. Start the backend Flask server by executing “**flask --app controller.py run**” on Windows or Linux.
6. Open a new terminal window and navigate to the optamo-coding-task /movies-search-engine-ui folder.
7. Run “**ng serve**” to build and serve the Angular application.
8. Open your browser and navigate to <https://localhost:4200> to start interacting with the application.
9. Please use any of the following credentials:
  - a. Username: john, password: test
  - b. Username: jane, password: admin

## Third-party libraries

### Angular:

@angular-devkit/build-angular@13.2.6

@angular-slider/ngx-slider@2.0.4

@angular/animations@13.2.7

@angular/cdk@13.3.9

@angular/cli@13.2.6

@angular/common@13.2.7

@angular/compiler-cli@13.2.7

@angular/compiler@13.2.7

@angular/core@13.2.7

@angular/forms@13.2.7

@angular/material@13.3.9

@angular/platform-browser-dynamic@13.2.7

@angular/platform-browser@13.2.7

@angular/router@13.2.7

@ngxs/store@3.8.0

@types/jasmine@3.10.7

@types/node@12.20.55

jasmine-core@4.0.1

karma-chrome-launcher@3.1.1

karma-coverage@2.1.1

karma-jasmine-html-reporter@1.7.0

karma-jasmine@4.0.2

karma@6.3.20

rxjs@7.5.7

tslib@2.5.0

typescript@4.5.5

zone.js@0.11.8

#### **Flask:**

gunicorn==20.1.0

Flask==2.2.3

Flask-Cors==3.0.10

PyJWT==2.6.0

requests==2.28.1

#### **Note:**

Please check “**Task\_1\_Data\_Processing\_Python.ipynb**” in `optamo-coding-task/movies-search-engine-api` for Task 1 – Data Processing (Python).