# Back end for CS673Proj

* The Back end is mainly a Flask Rest API
* The api\_module will handle all the call from the client side
* The file structure is represented as follows :

│   ├── api\_module

│   │   ├── \_\_init\_\_.py

│   │   ├── chat\_controller.py

│   │   ├── company\_controller.py

│   │   ├── employee\_controller.py

│   │   ├── helpers.py

│   │   ├── issue\_controller.py

│   │   ├── models.py

│   │   ├── project\_controller.py

│   │   ├── role\_controller.py

│   │   ├── sprint\_controller.py

│   │   ├── task\_controller.py

│   │   ├── team\_controller.py

│   │   └── user\_controllers.py

│   └── templates

│   ├── 404.html

│   └── docstring.html

├── config.py

├── local

│   └── db

│   └── app.db

├── readme.md

├── run.py

├── static

└── tests

├── payload.txt

└── pmp\_backend.postman\_collection.json

* Run a development server using the run.py
  + GET : <http://0.0.0.0:5005/api/>

# Requirements :

* Python 3.6
  + flask
  + SQLAlchemy
  + jwt
  + flask\_sqlalchemy
  + flask\_cors
  + flask\_socketio

# Run :

* python run.py

# PmpFrontend

## Project Structure

.

├── admin

│ ├── admin.component.html

│ ├── admin.component.scss

│ ├── admin.component.spec.ts

│ └── admin.component.ts

├── app-routing.module.ts

├── app.component.html

├── app.component.scss

├── app.component.spec.ts

├── app.component.ts

├── app.module.ts

├── auth

│ ├── auth.component.html

│ ├── auth.component.scss

│ ├── auth.component.spec.ts

│ └── auth.component.ts

├── chats

│ ├── chats.component.html

│ ├── chats.component.scss

│ ├── chats.component.spec.ts

│ └── chats.component.ts

├── guard

│ ├── auth.guard.spec.ts

│ └── auth.guard.ts

├── home

│ ├── home.component.html

│ ├── home.component.scss

│ ├── home.component.spec.ts

│ └── home.component.ts

├── issues

│ ├── font-awesome.min.css

│ ├── issues.component.html

│ ├── issues.component.scss

│ ├── issues.component.spec.ts

│ └── issues.component.ts

├── models

│ └── issue.ts

├── nav

│ ├── nav.component.html

│ ├── nav.component.scss

│ ├── nav.component.spec.ts

│ └── nav.component.ts

├── services

│ ├── auth.service.spec.ts

│ ├── auth.service.ts

│ ├── issuetracker.service.spec.ts

│ └── issuetracker.service.ts

└── sprints

├── sprints.component.html

├── sprints.component.scss

├── sprints.component.spec.ts

└── sprints.component.ts

This project was generated with [Angular CLI](https://github.com/angular/angular-cli) version 7.3.2.

## Development server

Run ng serve for a dev server. Navigate to http://localhost:4200/. The app will automatically reload if you change any of the source files.

## Code scaffolding

Run ng generate component component-name to generate a new component. You can also use ng generate directive|pipe|service|class|guard|interface|enum|module.

## Build

Run ng build to build the project. The build artifacts will be stored in the dist/ directory. Use the --prod flag for a production build.

## Running unit tests

Run ng test to execute the unit tests via [Karma](https://karma-runner.github.io/).

## Running end-to-end tests

Run ng e2e to execute the end-to-end tests via [Protractor](http://www.protractortest.org/).

## Further help

To get more help on the Angular CLI use ng help or go check out the [Angular CLI README](https://github.com/angular/angular-cli/blob/master/README.md).