Angular Social (MEAN)



|  |  |
| --- | --- |
| LAYER | TECHNOLOGIES |
| CLIENT SIDE | Angular + NGRX + Bootstrap |
| SERVER SIDE | NODE + EXPRESS + MONGO DB |

Folder Structure

Angular-Social

-> client

-> Client Code

-> Server Code

Client Configuration

-> Create an Angular Application "client" -> ng new client

Angular Modules

Users , profiles , posts

ng generate module profiles --route profiles --module app.module

ng generate module posts --route posts --module app.module

ng generate module users --route users --module app.module

-> each module have its own things

components ,

services ,

interceptors ,

guards ,

models ,

actions ,

reducers ,

effects

-> Users Module :

components : login , register , profile

services : user.service

-> Posts module :

components : posts , comments

services : post.service

Server Configuration (Express + MongoDB)

|  |  |
| --- | --- |
| Modules | Details |
| Modules | bcryptjs ,  cors ,  dotenv ,  express ,  express-validator ,  gravatar ,  jsonwebtoken,  mongoose  axios |
| npm install bcryptjs cors dotenv express express-validator gravatar jsonwebtoken mongoose axios | |

REST API Configuration

Routers : profileRouter , userRouter , postRouter

User Router

|  |  |
| --- | --- |
| 1. Register a User |  |
| URL | api/users/register |
| Fields | Name , email , password |
| Method | POST |
| Access | PUBLIC |

|  |  |
| --- | --- |
| 2. Login a User |  |
| URL | api/users/login |
| Fields | email , password |
| Method | POST |
| Access | PUBLIC |

|  |  |
| --- | --- |
| 3. Get User Info |  |
| URL | api/user/ |
| Fields | No-fields |
| Method | GET |
| Access | PRIVATE |

Profile Router

|  |  |
| --- | --- |
| 1. Get a profile |  |
| URL | api/profiles/me |
| Fields | No-fields |
| Method | GET |
| Access | PRIVATE |

|  |  |
| --- | --- |
| 2. Create a Profile |  |
| URL | /api/profiles/ |
| Fields | company , website , location , designation , skills , bio , githubUsername, youtube , facebook , twitter , linkedin , instagram |
| Method | POST |
| Access | PRIVATE |

|  |  |
| --- | --- |
| 3. Update a Profile |  |
| URL | /api/profiles/ |
| Fields | company , website , location , designation , skills , bio , githubUsername, youtube , facebook , twitter , linkedin , instagram |
| Method | PUT |
| Access | PRIVATE |

|  |  |
| --- | --- |
| 4. Get a Profile of a User |  |
| URL | /api/profiles/users/:user\_id |
| Fields | No-fields |
| Method | GET |
| Access | PUBLIC |

|  |  |
| --- | --- |
| 5. Delete a Profile , User , Posts |  |
| URL | /api/profiles/users/:user\_id |
| Fields | No-fields |
| Method | DELETE |
| Access | PRIVATE |

|  |  |
| --- | --- |
| 6. Update an experience of a Profile |  |
| URL | /api/profiles/experience/ |
| Fields | title , company , location , from , to , current , description |
| Method | PUT |
| Access | PRIVATE |

|  |  |
| --- | --- |
| 7. Delete an experience of a Profile |  |
| URL | /api/profiles/experience/:exp\_id |
| Fields | No-fields |
| Method | DELETE |
| Access | PRIVATE |

|  |  |
| --- | --- |
| 8. Update an education of a Profile |  |
| URL | /api/profiles/education |
| Fields | school , degree , fieldOfStudy , from , to , current , description |
| Method | PUT |
| Access | PRIVATE |

|  |  |
| --- | --- |
| 9. Delete an education of a Profile |  |
| URL | /api/profiles/education/:edu\_id |
| Fields | No-fields |
| Method | DELETE |
| Access | PRIVATE |

POSTS Router

|  |  |
| --- | --- |
| 1. Create a POST |  |
| URL | /api/posts |
| Fields | User , text , name , avatar |
| Method | POST |
| Access | PRIVATE |

|  |  |
| --- | --- |
| 2. GET all Posts |  |
| URL | /api/posts/ |
| Fields | No fields |
| Method | GET |
| Access | PRIVATE |

|  |  |
| --- | --- |
| 3. GET a post with post id |  |
| URL | /api/posts/:post\_id |
| Fields | No fields |
| Method | GET |
| Access | PRIVATE |

|  |  |
| --- | --- |
| 4. DELETE a post |  |
| URL | /api/posts/:post\_id |
| Fields | No fields |
| Method | DELETE |
| Access | PRIVATE |

|  |  |
| --- | --- |
| 5. LIKE a post |  |
| URL | /api/posts/like/:post\_id |
| Fields | No fields |
| Method | PUT |
| Access | PRIVATE |

|  |  |
| --- | --- |
| 6. Un-LIKE a post |  |
| URL | /api/posts/unlike/:post\_id |
| Fields | No fields |
| Method | PUT |
| Access | PRIVATE |

|  |  |
| --- | --- |
| 7. CREATE Comment to a post |  |
| URL | /api/posts/comment/:post\_id |
| Fields | Text -> user , text , name , avatar |
| Method | POST |
| Access | PRIVATE |

|  |  |
| --- | --- |
| 8. DELETE a Comment of a post |  |
| URL | /api/posts/comments/:post\_id/:comment\_id |
| Fields | No-fields |
| Method | DELETE |
| Access | PRIVATE |

Database Configuration

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Database Name | angular-social-02-aglr-evng |  |  |
| Tables | Users | Posts | Profiles |
| Users Table | Name, email , password , isAdmin , avatar , created | | |
| Profiles Table | User , company , website , location , designation , skills , bio , githubusername , experience[  {title , company , location , from , to ,current , description }  ] , education [  { school ,degree , fieldOfStudy , from , to , current , description }  ],  social : { youtube , twitter , facebook , linkedin , Instagram },  created | | |
| Posts Table | User , text , name , avatar , likes [  { user}  ],  Comments : [  { user , text , name , avatar , date }  ],  created | | |