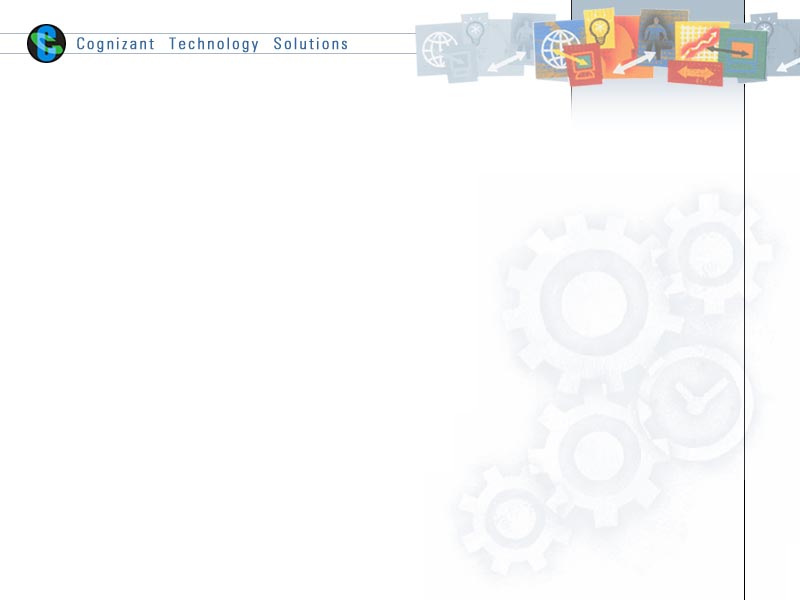
**Angular2 – HomeTutor.in**

Requirement Document

**Version No. 0.2**



Contents

[1. Case study Description 2](#_Toc493693582)

[2. Low level design/Flow Diagram: 3](#_Toc493693583)

[3. High Level Design/ Flow Diagram: 3](#_Toc493693584)

[4. Code Skeleton: 4](#_Toc493693585)

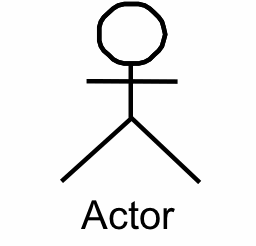
[5. Instructions to Associates: 4](#_Toc493693586)

[6. Case study Requirements 7](#_Toc493693587)

# Case study Description

Hey buddy, how are you enjoying the Android or Front-end class and what are the concepts that you learned today? Has he taught anything which is too difficult? Did he also give you some assignments which you think is not quite easy? If so, here is something that can be explored. The application HOMETUTOR.COM is something to help people handle such situations.

# Low level design/Flow Diagram:



# High Level Design/ Flow Diagram:

Login or sign up to tutor application

Select course and click on find my tutor

Enter problem description, select tutor name, and submit query

***Achieve all the below requirements using the above flow diagram.***

# Code Skeleton:

Code skeleton contains the basic libraries and base files of the skills, so that the exercise

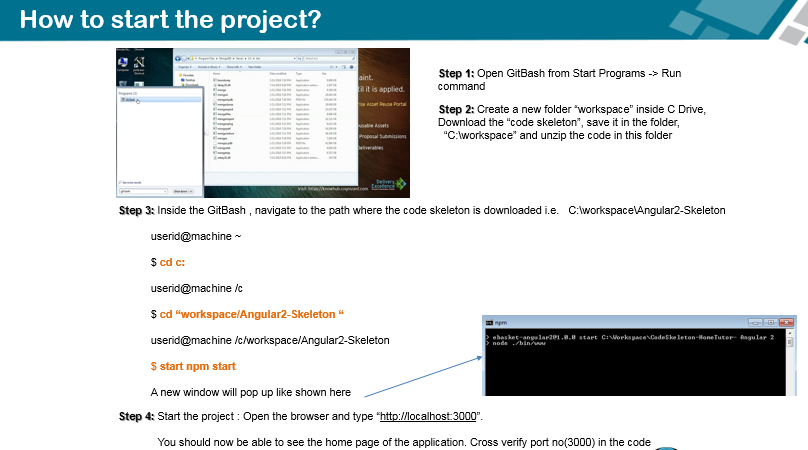
Solutions can be built on top of it seamlessly.

# Instructions to Associates:

The code skeleton contains a full-fledged working node.js API for all the required GET and POST HTTP requests from the UI. It also contains the basic libraries and base files of the front-end so that the solutions can be built on top of it seamlessly.

* Please download the project code skeleton. The code skeleton contains a full-fledged working node.js API for all the required GET and POST Ajax calls from the UI. It also contains the basic libraries and base files of the front end, so that the exercise solutions can be built on top of it seamlessly
* Note :
  + Necessary software are available @ VDI Desktop -> Academy Digital Folder
  + Mongodb is available @ “C:\Program Files\MongoDB\Server\3.3\bin“
  + Use the GitBash instead of the command prompt as given in the case study requirement document.
* Follow the below steps as provided in the screenshot to start the mongodb, node server and the project application

##### 

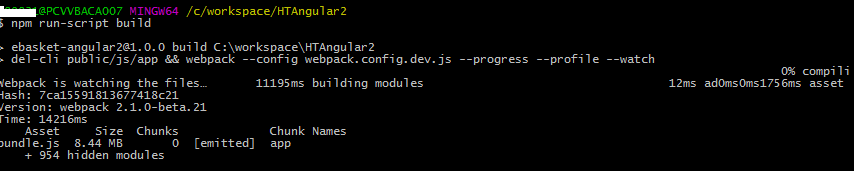


***Please note: After each code update you might have to refresh the browser for changes to reflect and correspondingly you can see the logs in the console***

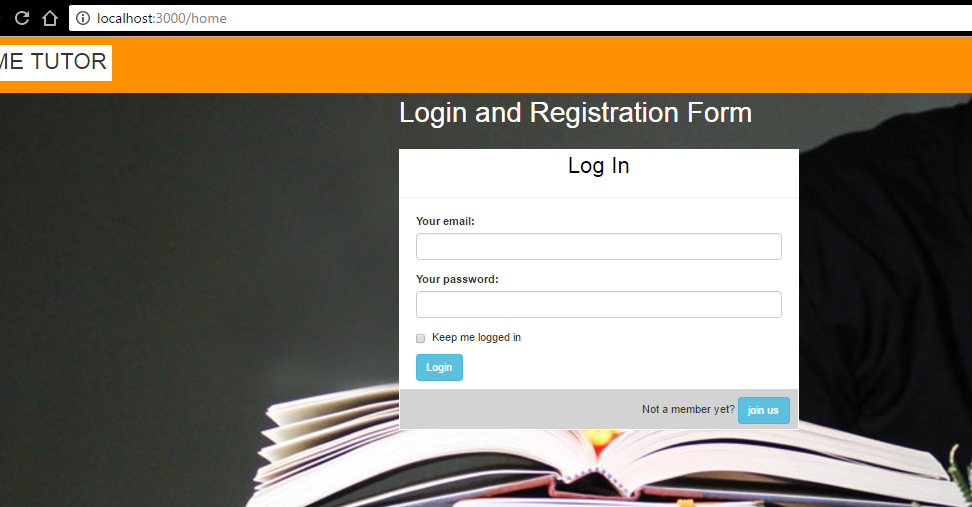
**Execute the below command for code changes to be built automatically:**

Execute the below command from project root folder. E.g: ”C:/Workspace/Angular2-Skeleton”

**$ npm run-script build**

******

Snapshot of Application Home page :



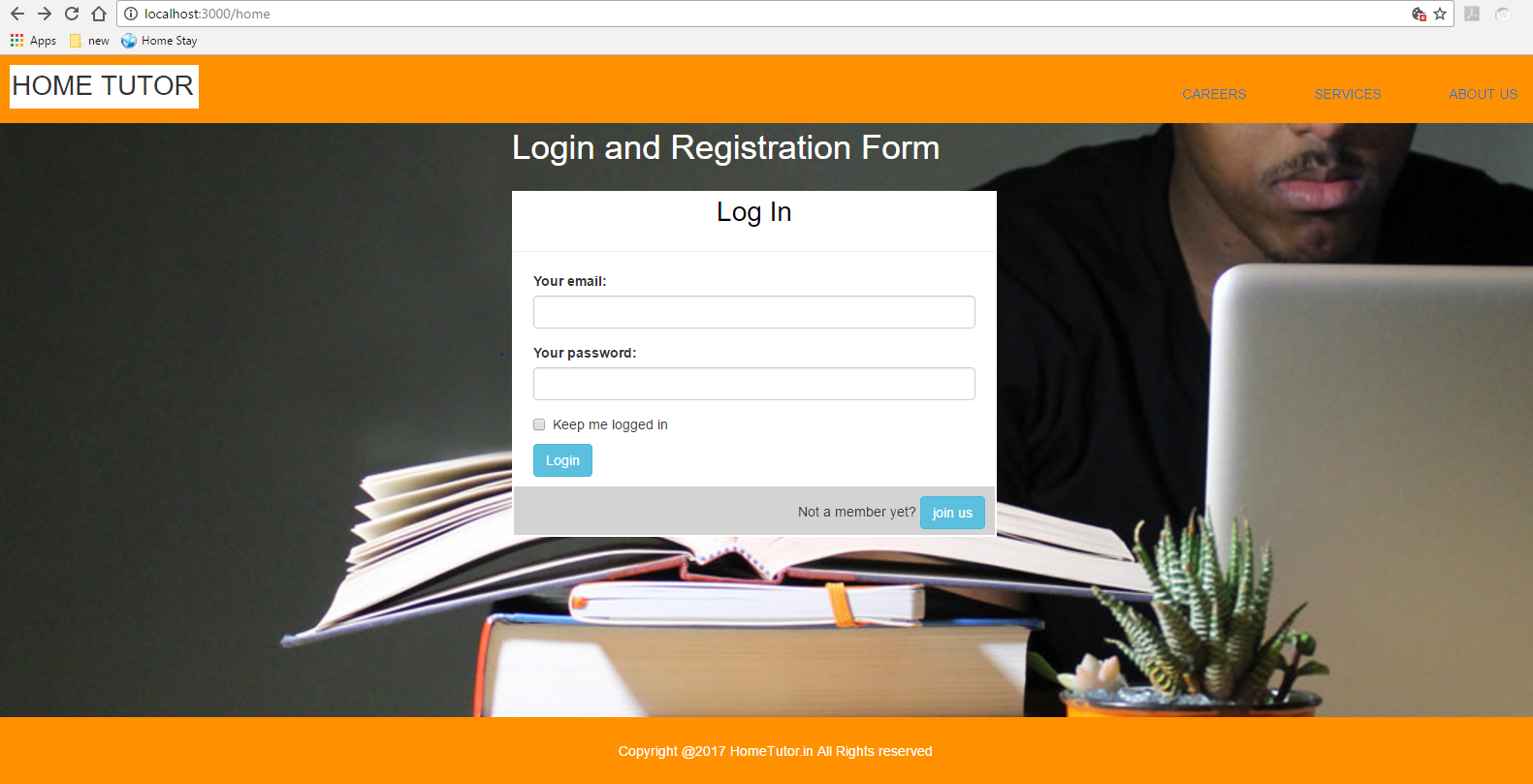
# Case study Requirements

**Requirement 1:**

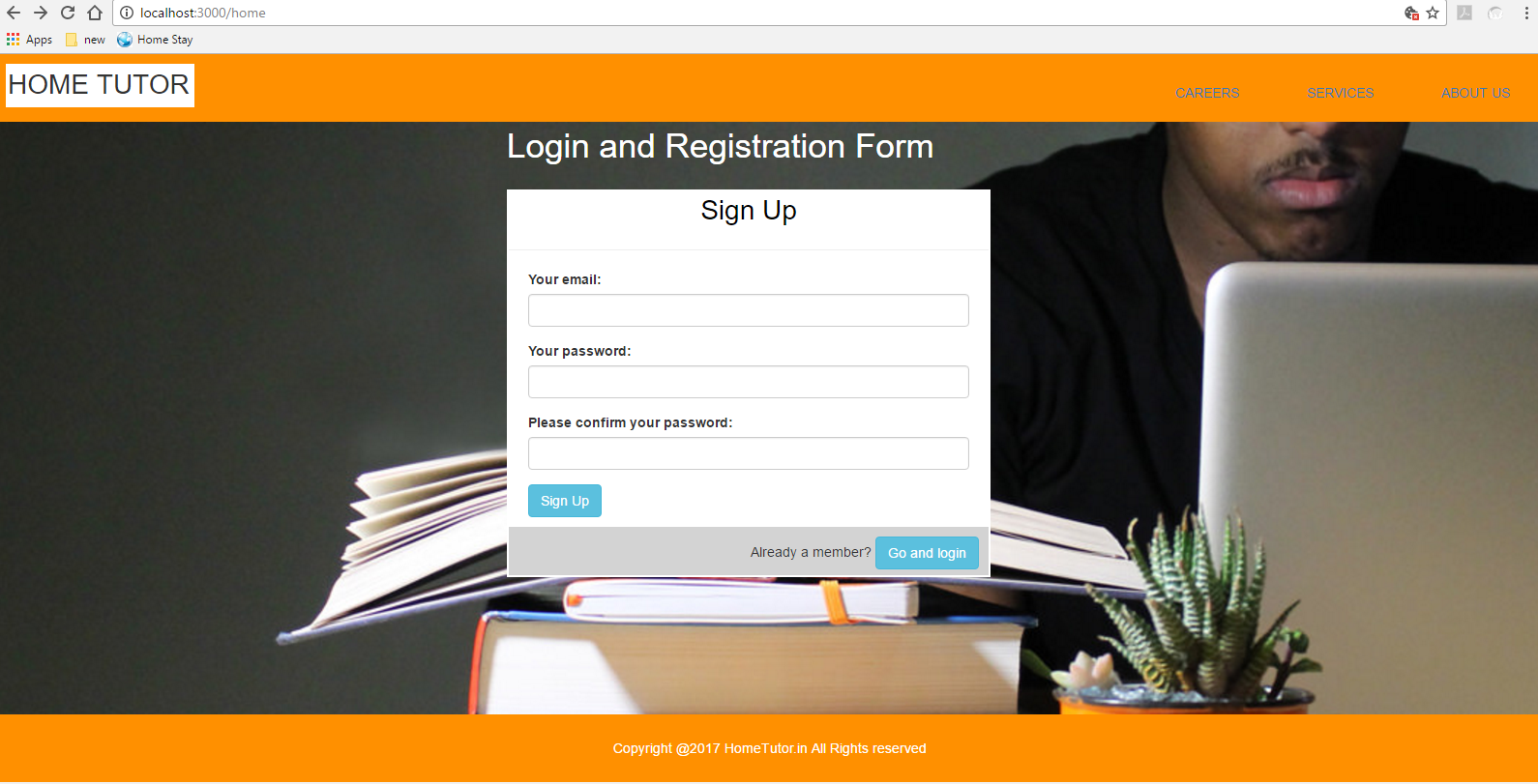
|  |  |
| --- | --- |
| Problem Statement | Design the home page with the customized header and footer and sign in/signup forms. All fields should be validated with proper error message  Sign in:   * Email * Password * Submit button * Not a member yet? Link to sign in the Sign up form * Email * Password * Confirm password * Submit button * Already member? Link to Sign in form |
| Design Consideration | |  |  |  |  | | --- | --- | --- | --- | | Field Name | Field Type | Restrictions | Mandatory | | Email | Input text | Valid email | Yes | | Password | Input password | None | Yes | | Confirm password | Input password | Should be same as password | Yes |   Make sure to have the app name in the header and copyright symbol in the footer.  Make sure you have placed HD as the background image.  Make sure the design of sign in and sign up form should be covered in the same layout. |
| Business Rules and Process | On the click of the Join us button, we should be able to get the sign up form.  On the click of the Go and Login button, we should be able to get the sign in form.   1. User enters email and password 2. Validate the forms 3. If forms are valid, make http requests to server 4. If request is successful, redirect user to the dashboard page 5. If any error is there, show it properly in UI |
| Skills and Features | Angular features used:   * @angular/core * @angular/http * @angular/router * @angular/forms * Rxjs   Forms, Dom Manipulation, Http, Routing and navigation, Component, Services, Pipes, NgIf, Event handling, Route Guards |
| Connected code in skeleton | \assets\app\home   1. [Home.component.ts-](http://Home.component.ts-) TODO1, TODO2, TODO3 2. Home.component.html 3. Data.service.ts - TODO4, TODO5, TODO6 |
| Time required | 1.5 hours |

**UI Design:** Build the following screens.

Home Screen



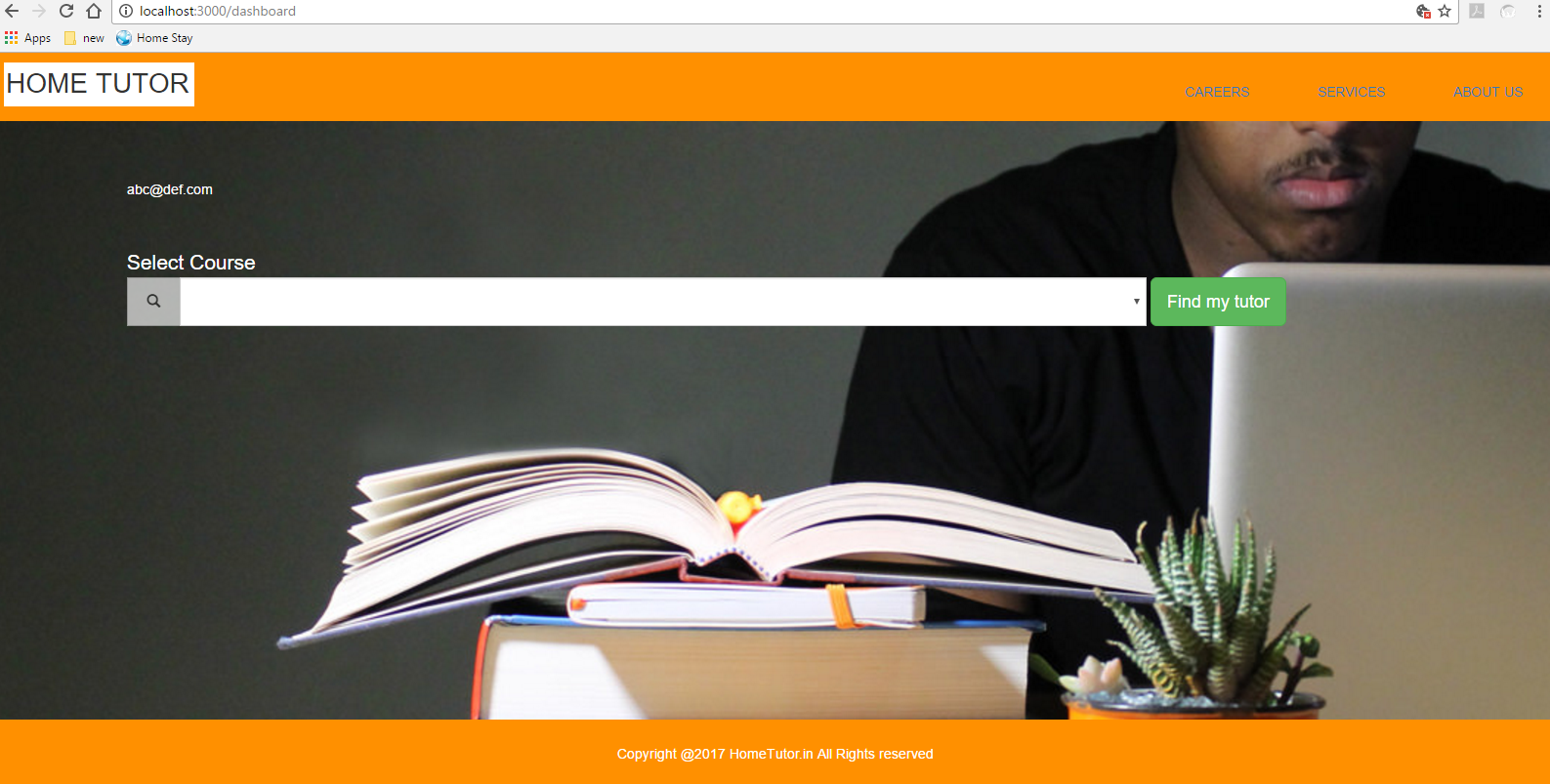
Registration form screen on the click of Join us button



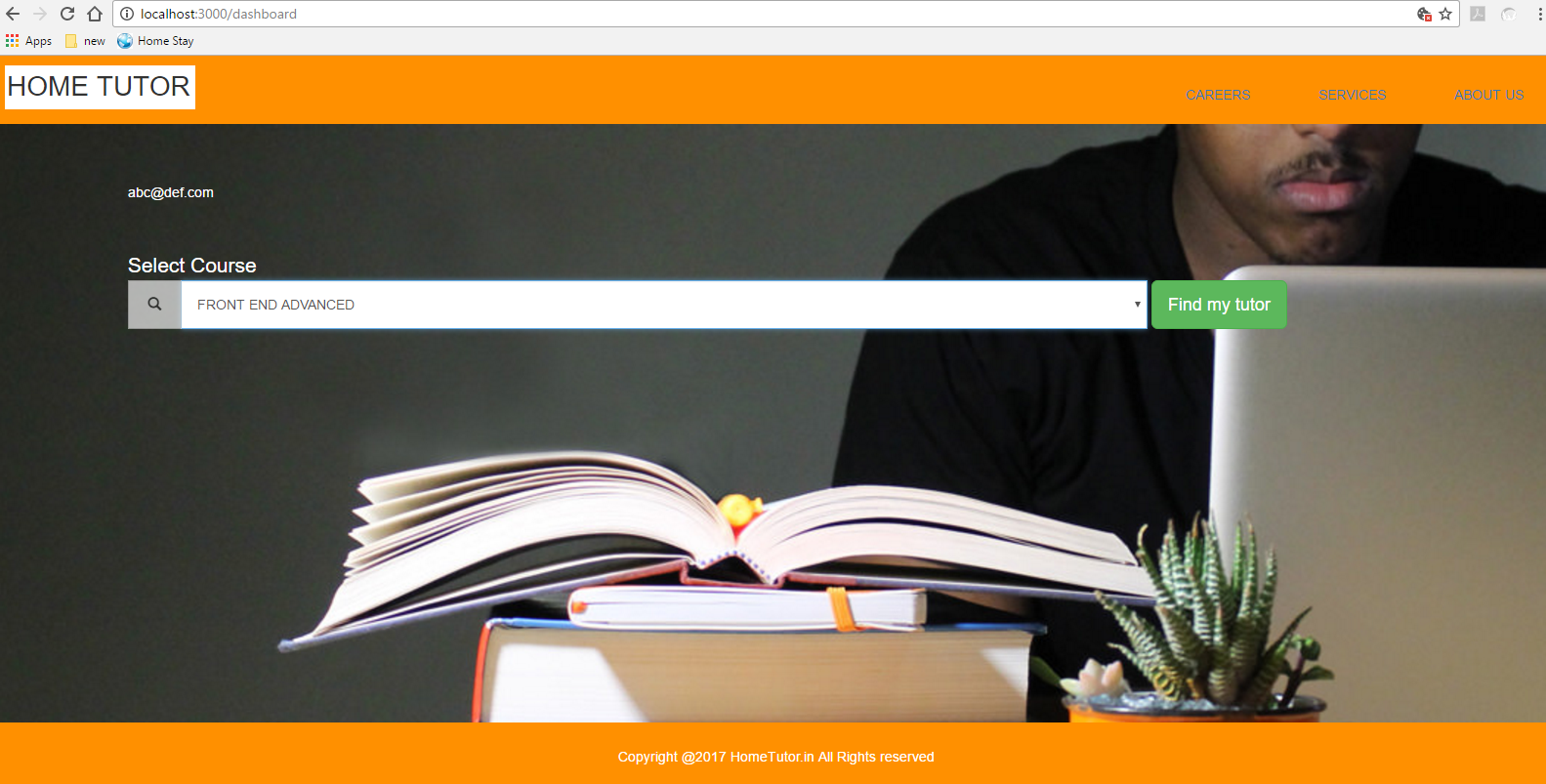
**Requirement 2:**

|  |  |
| --- | --- |
| Problem Statement | Design a search page which can find the tutor for a particular subject selected.   * Drop down containing list of courses – Android Basics, Front End Advance * “find my tutor” button |
| Design Consideration | Should have a search bar drop-down which should be able to select the items  Should have the button which can place search queries   |  |  |  |  | | --- | --- | --- | --- | | Field Name | Field Type | Restrictions | Mandatory | | Select course | Input drop down | none | Yes | |
| Business Rules and Process | 1. Show the email of the logged in user on the page. 2. Select one of the courses from the drop down. 3. On click on “find my tutor” button, make an Http get request along with mentor name as parameter. |
| Skills and Features | Angular features used:   * @angular/core * @angular/http * @angular/router * @angular/forms * Rxjs   Forms, Dom Manipulation, Http, Routing and navigation, Component, Services, NgIf, Event handling |
| Connected code in skeleton | /assets/app/home   1. Data.service.ts – TODO7, TODO9   /assets/app/dashboard   1. Dashboard.component.ts – TODO8 |
| Time required | .5 hours |

ON CLICK OF THE SIGN IN BUTTON, WE SHOULD BE ABLE TO GET THE BELOW SCREEN.



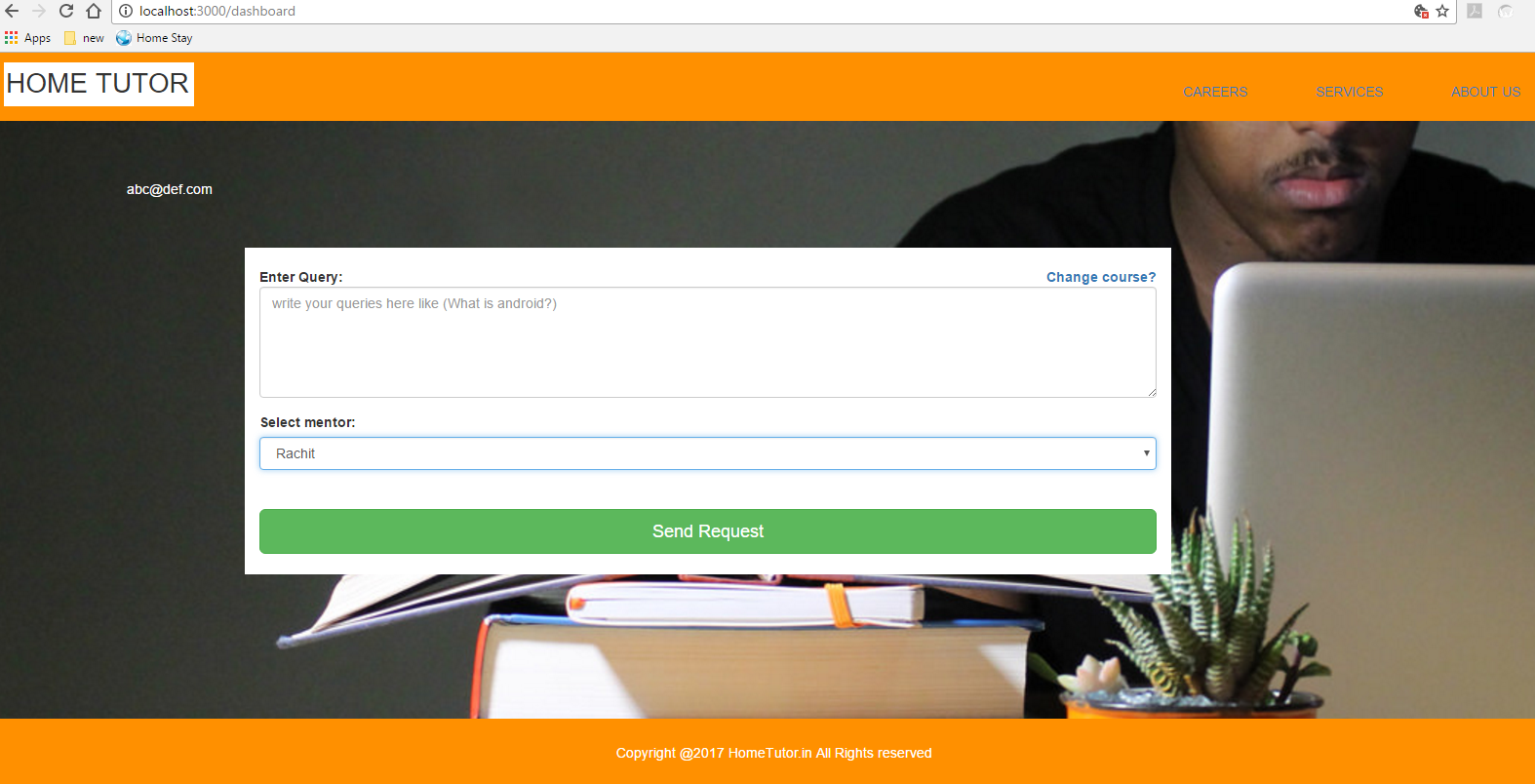
ON SELECTING THE DROP-DOWN, WE SHOULD GET THE BELOW SCREEN.



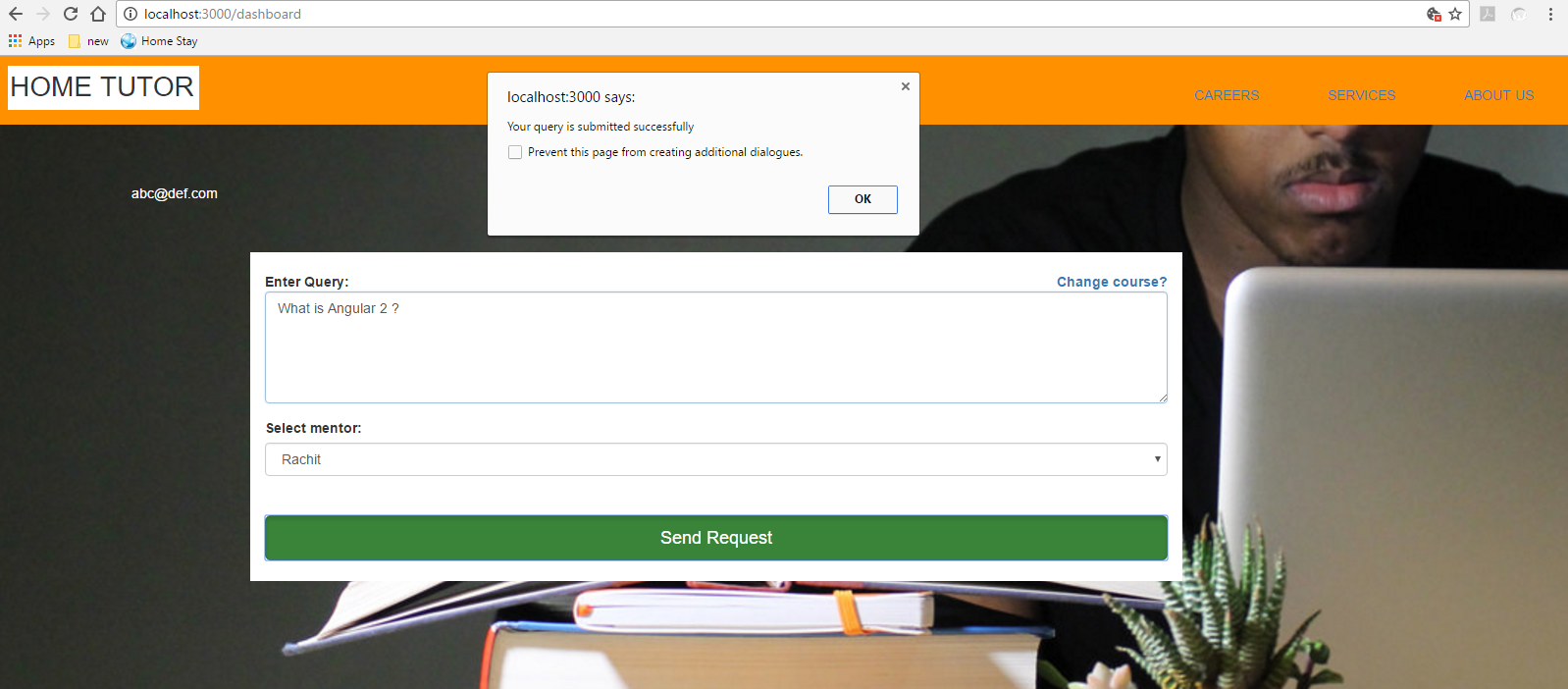
**Requirement 3:**

|  |  |
| --- | --- |
| Problem Statement | Design the page where user can ask query   * Text area to write query * Drop-down having list of mentors/tutors * Submit button to send queries |
| Design Consideration | Should have the proper design which should include drop-down and the submit button  Should develop the pop up message box as the query response   |  |  |  |  | | --- | --- | --- | --- | | Field Name | Field Type | Restrictions | Mandatory | | Enter Query | Textarea | None | Yes | | Mentor name | Select option | Single select | Yes | |
| Business Rules and Process | 1. Show the query form to the user. 2. Bind list of mentors in the drop-down. 3. User enters the query and selects one of the mentors from the drop-down. 4. Click on submit query button to send query. 5. Show pop up message “” when query is successfully submitted. |
| Skills and Features | Angular features used:   * @angular/core * @angular/http * @angular/router * @angular/forms * Rxjs   Forms, Dom Manipulation, Http, Component, Services, NgIf, Event handling |
| Connected code in skeleton | /assets/app/home   1. Data.service.ts – TODO11   /assets/app/dashboard   1. Dashboard.component.ts – TODO10 |
| Time required | 1 hour |

ON SELECTING THE PARTICULAR SELECTED ITEM CATEGORY FROM THE REQUIREMENT 02, WE SHOULD GET THE BELOW PAGE.



ON HITTING THE SEND REQUEST BUTTON, WE SHOULD GET THE BELOW PAGE.



**API Endpoints:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field/ Purpose** | **HTTP Verb** | **Url** | **Mandatory**  **Params** |
| Sign In | POST | http://localhost:3000/users/signin  example:  <http://localhost:3000/users/signin>  {  email : ‘abc@def.com’,  password: 123456  }  Result: 200  {  "message":"success",  "data": {  "email":"abc@def.com"  }  } | 1. email 2. password |
| Sign Up | POST | <http://localhost:3000/users/signup>  example:  <http://localhost:3000/users/signup>  {  email : ‘abc@def.com’,  password: 123456  }  Result: 201  {  "message":”user created",  "data": {  "email”: “abc@abc.com”  }  } | 1. email 2. password |
| Search Tutors | GET | <http://localhost:3000/mentors>  example:  <http://localhost:3000/mentors?course=ANDROID%20BASICS>  result: 200  {"message":"success","data":["Pushpa","John"]} | 1. course |
| Submit Query | POST | <http://localhost:3000/query>  example:  <http://localhost:3000/query>  {  query: "what is angular 2?",  mentor: "Rachit",  email: [abc@def.com](mailto:abc@def.com)  }  Result: 201  {  "message":”Query Registered",  "data": {  "email”: “abc@abc.com”  “query”: “what is angular 2?”  “mentor”: “Rachit”  }  } | 1. query 2. mentor 3. email |

======================End of the document====================