CS5551-ADVANCED SOFTWARE ENGINEERING

PROJECT INCREMENT-1 REPORT

TITLE: A "GitHub" for Deep Learning Models: Online Host, Visualization, and Deployment.

TEAM NAME: CODE BREAKERS

TEAM ID: 5

TEAM MEMBERS:

1. TEJA DEVARAPALLI
2. RAHUL DHAMERLA
3. GOPI CHAND BODEPUDI
4. VASUDEVA MADALA

INTRODUCTION:

This project aims to create a hosting service similar to GitHub that allows users to host their trained
models, share them with the public, search for models, rank the models, allow model download, and
online testing and deployment.

PROJECT GOAL AND OBJECTIVES:

We developed a web application similar to 'GitHub' for users where they

can host their models and

providing access for them to download the models and rate them accordingly.

MOTIVATION:

With the advances in deep learning applications, sharing the trained models online becomes a critical

component of the overall deep learning lifecycle. Currently, trained models are share on private websites

or dumped into storage services like GitHub, which cannot visualize the model and its metadata, test it

online, or deploy it online.

SYSTEM REQUIREMENTS:

• A hosting website. Users can create accounts and have profiles, similar to a social network, or GitHub.

• Users can upload their trained models to the website.

• Users can "like" or "vote" up and down for models, comment on them and submit issues on a model (or report a model)

• Models can be downloaded from the website.

• Users has dashboards about their models, views, ratings, etc.

PROJECT PLAN USING ZEN HUB TOOLS:

Schedule for four increments:

Milestones

New Milestone

Increment-1
Start **Feb 8, 2019** Due by **Feb 26, 2019** Duration: **19 days**

See this Milestone on the Board

Edit Milestone Close Milestone Delete Milestone

Increment-2
Start **Feb 8, 2019** Due by **Mar 19, 2019** Duration: **a month**

See this Milestone on the Board

Edit Milestone Close Milestone Delete Milestone

Increment-3
Start **Feb 8, 2019** Due by **Apr 16, 2019** Duration: **2 months**

See this Milestone on the Board

Edit Milestone Close Milestone Delete Milestone

Increment-4
Start **Feb 8, 2019** Due by **May 3, 2019** Duration: **3 months**

See this Milestone on the Board

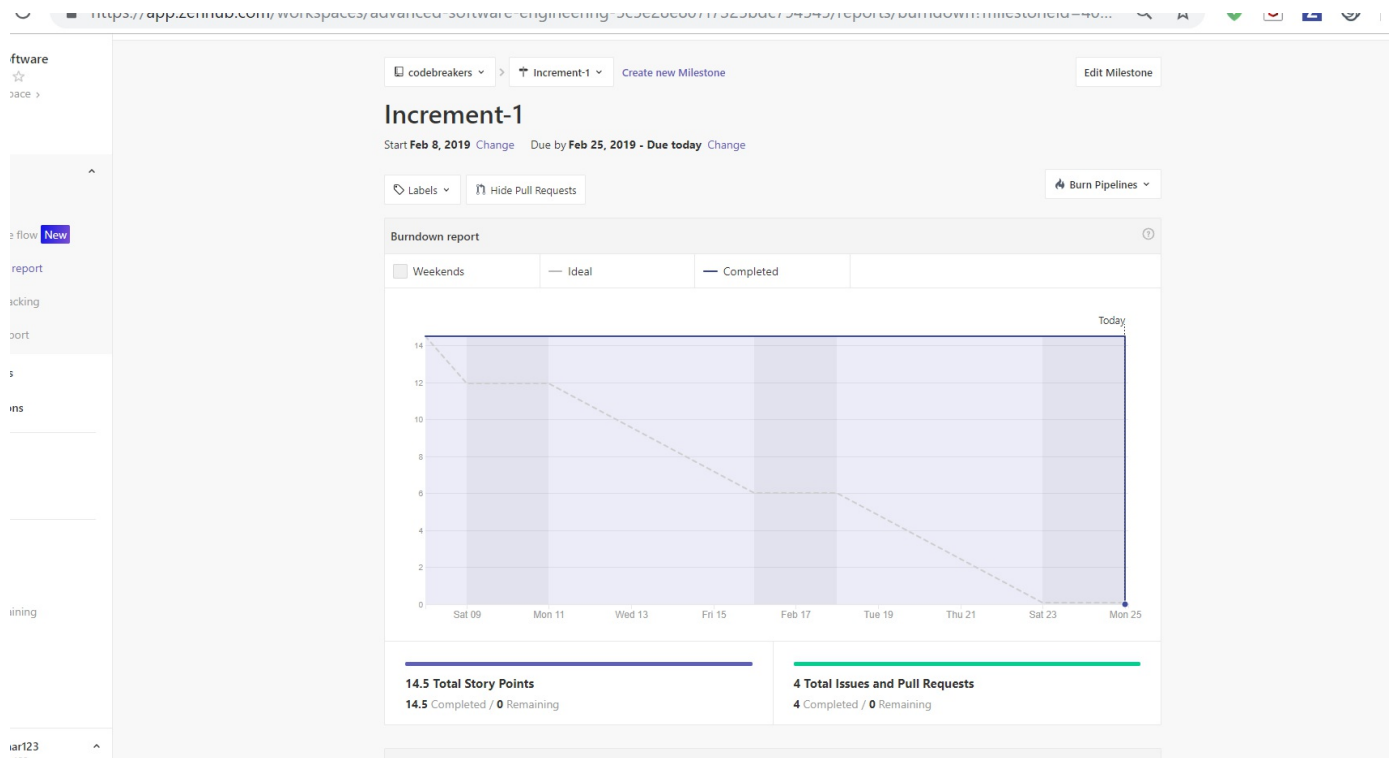Edit Milestone Close Milestone Delete Milestone

Project Timeliness:

The below are the issues raised during the increment-1 and moved them to closed state once the

individual task is done.

BURNDOWN CHART:

1. Design and Architecture

2. Login, Register User Interface

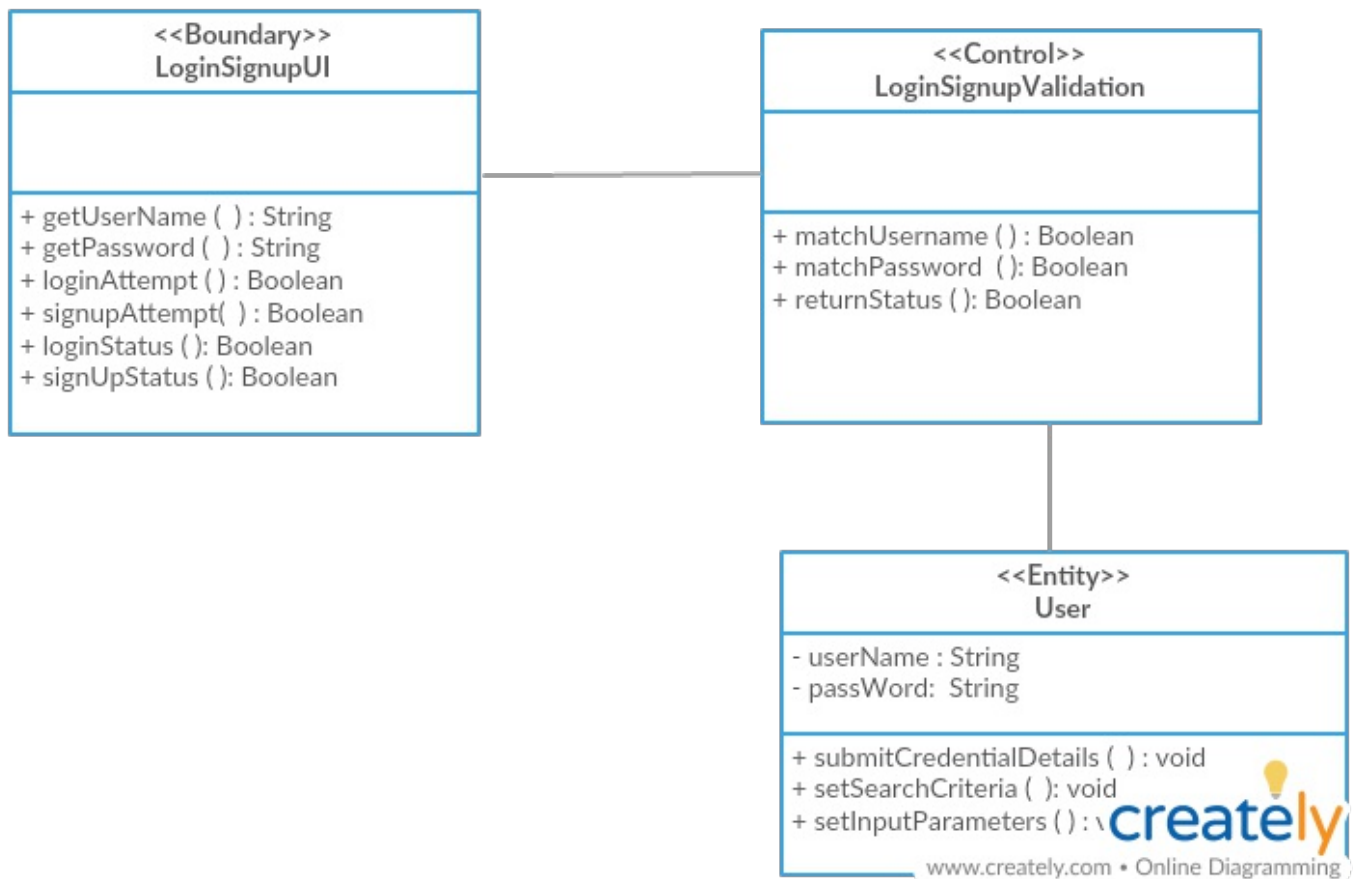3. UML Diagrams

FIRST INCREMENT REPORT:

INTRODUCTION:

A web application that avails users to register with valid credentials. We have setup Firebase
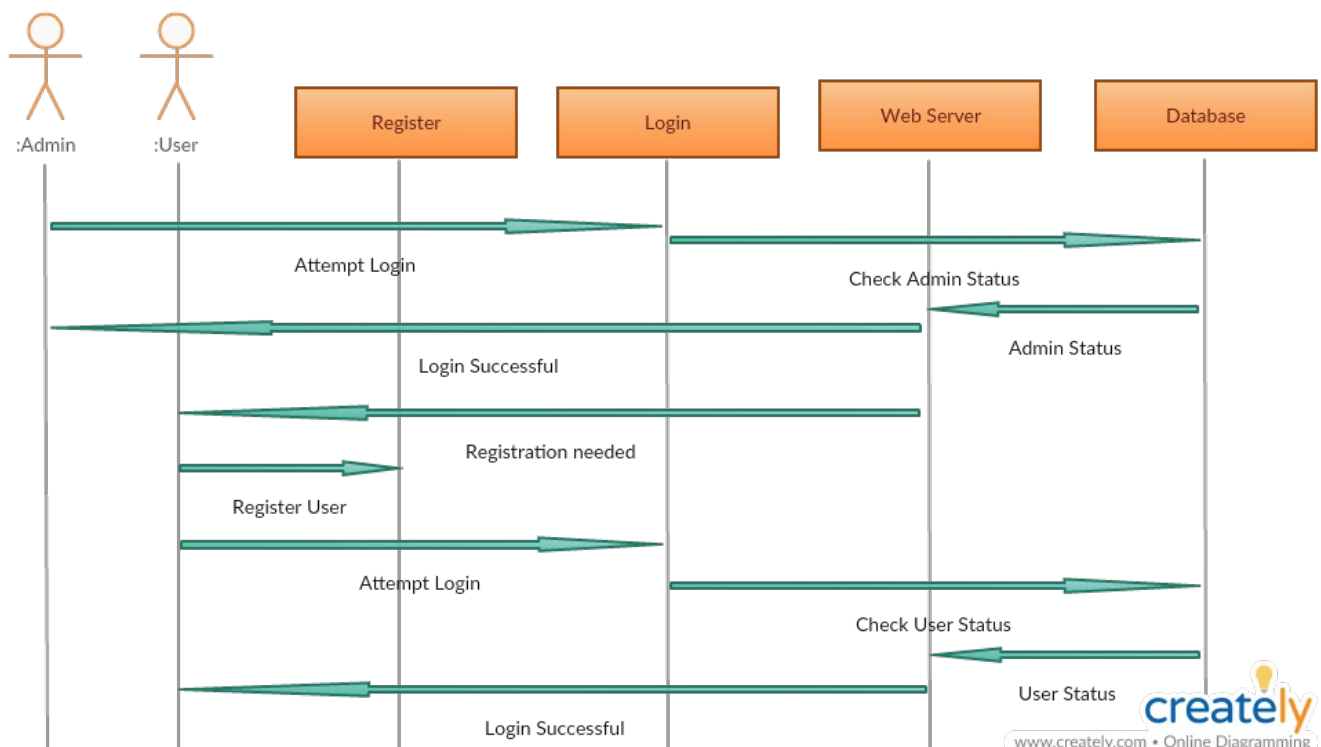Authentication and Firebase DB that stores credentials of the users.

TECHNOLOGIES USED:

☐ HTML5, CSS3

☐ BOOTSTRAP

☐ ANGULAR7

☐ FIRE BASE AUTHENTICATION

☐ FIRE BASE STORAGE

UML DIAGRAMS:

1.Class Diagram:

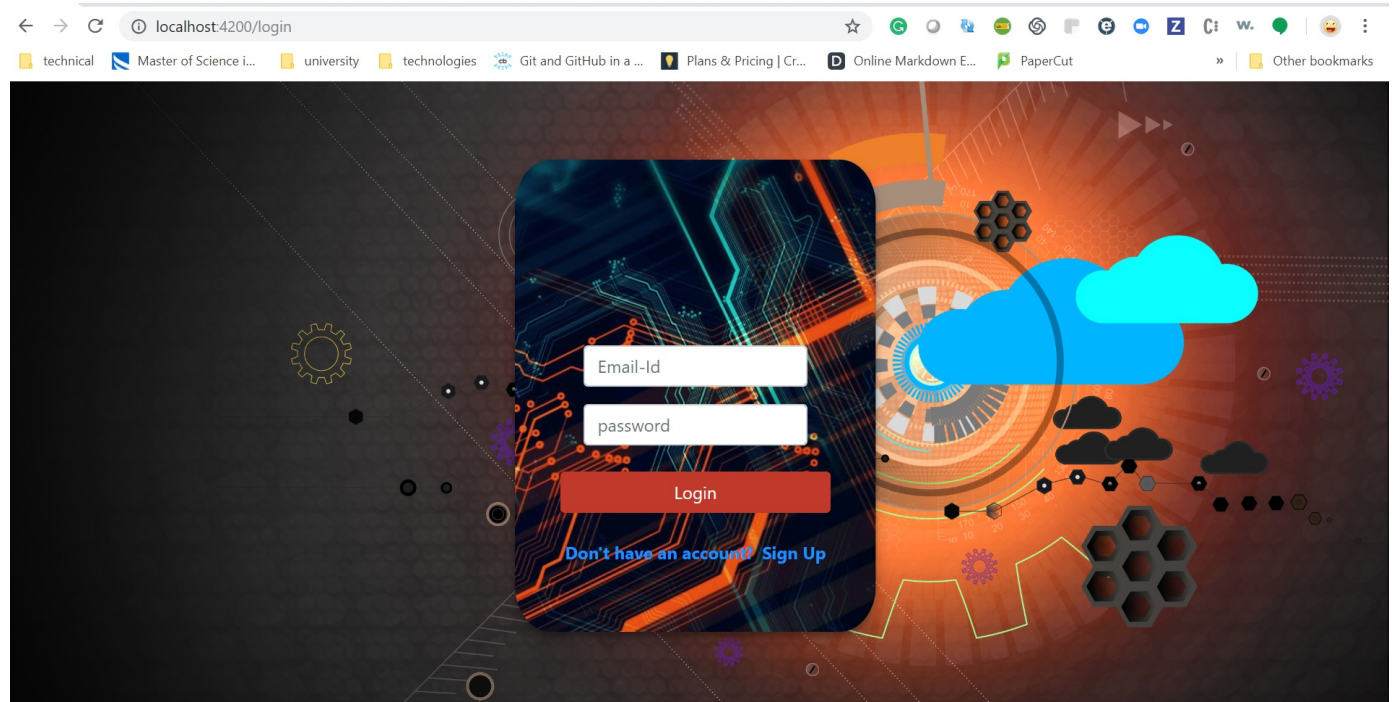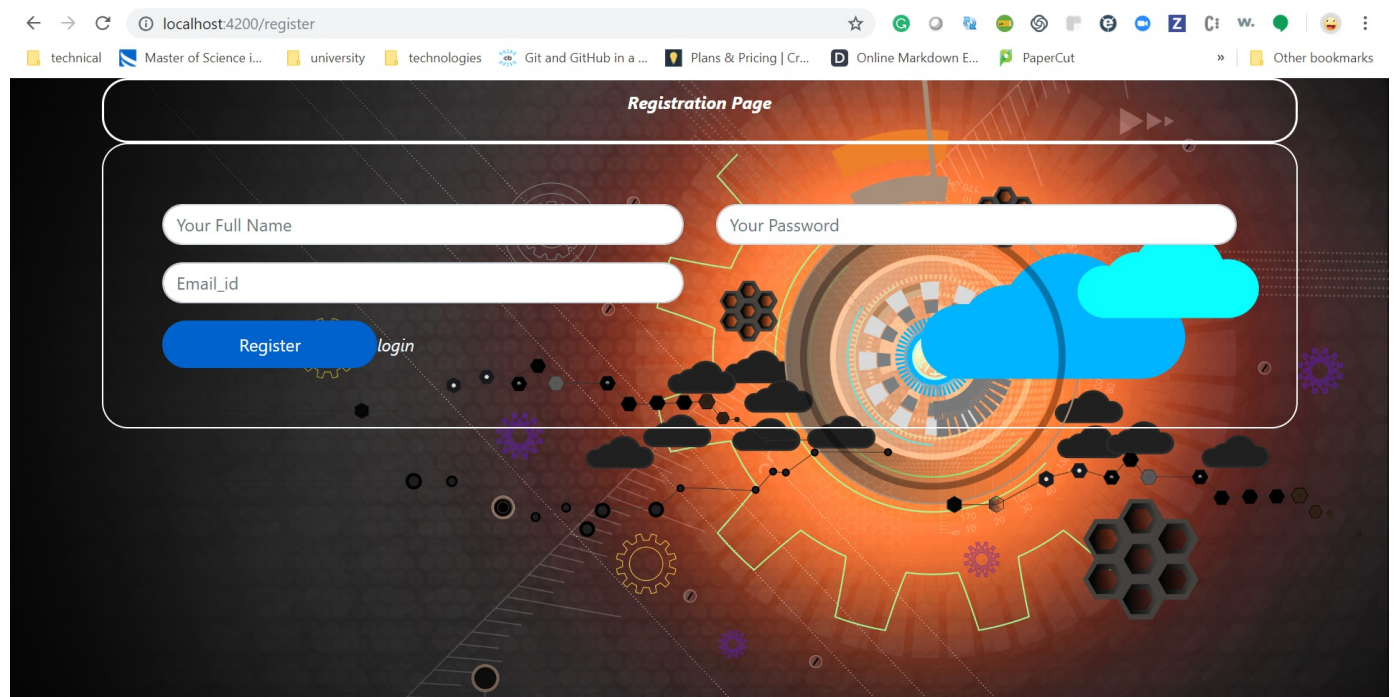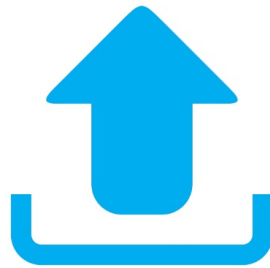# 2.Sequence Diagram:

OUTPUT:

Login page

Register page

Home page

Upload your files here

Home.html


```html
<body>
<ul style="padding-left: 280px; background-color: lightslategray; " >
  <li><a class="active" href="#home">Pull Requests</a></li>
  <li><a href="#news">Issues</a></li>
  <li><a href="#contact">Market Place</a></li>
  <li><a href="#about">Explore</a></li>
  <li><a href="#about" style="margin-left: 460px;">Profile</a></li>
</ul>
<div class="item">
  <img src="https://stmastering.co.uk/wp-content/uploads/2016/09/Very-Basic-Upload-icon.jpg" style=" width: 303px;marg
</div>
</body>

<div id="wrapper"  [ngClass]="ref.click === true? 'toggled' :''" >
  <div class="overlay" [ngStyle]="{disply:ref.click === true? 'block' :'none'}" ></div>
  <nav class="navbar navbar-inverse navbar-fixed-top" id="sidebar-wrapper" role="navigation">
    <ul class="nav sidebar-nav" style="height: 100%;background-color: lightslategray;">
      <li class="sidebar-brand">

      </li>
      <li>
        <a href="#">My Modules</a>
      </li>
      <li>
        <a href="#">Modules</a>
      </li>
```

Home.ts

```
    import { Component, OnInit } from '@angular/core';
    import {FireauthService} from '../fireauth.service';
    import {Router} from '@angular/router';

    @Component({
        selector: 'app-home',
        templateUrl: './home.component.html',
        styleUrls: ['./home.component.css']
    })
    export class HomeComponent implements OnInit {

        constructor(public router: Router) { }

        ngOnInit() {

            signout()
            {

                this.router.navigate( commands: ['']);
            }
        }
    }
```

Login.html

```html
<div class="user_card">
    <div class="d-flex justify-content-center form_container">
        <form>
            <div class="input-group mb-3">
                <input type="text"  class="form-control" placeholder="Email-Id" type="email" [(ngModel)]="email" name="email
            </div>
            <div class="input-group mb-2">
                <input type="password"  class="form-control"  placeholder="password" [(ngModel)]="password" name="password"/
            </div>
        </form>
    </div>
    <div class="d-flex justify-content-center mt-3 login_container">
        <button type="button" class="btn login_btn" (click)="login()">Login</button>
    </div>
    <div class="mt-4">
        <div class="d-flex justify-content-center links">
            <p style="..."><b>Don't have an account?</b></p>
            <nav class="ml-2"><a style="..." routerLink="/register"><b>Sign Up</b></a></nav>
        </div>
    </div>
</div>
</div>
</div>
```

div.container.h-100 > div.d-flex.justify-content-center.h-100

Login.ts

```typescript
import { Component, OnInit } from '@angular/core';
import { FireauthService} from '../fireauth.service';
import {Router} from '@angular/router';

@Component({
    selector: 'app-login',
    templateUrl: './login.component.html',
    styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {
public email;
public password;
public eMessage;
public sMessage;
    constructor(public auth: FireauthService, public router: Router) { }

    ngOnInit() {
    }
    login() {
        this.auth.firelogin(this.email, this.password).then( onfulfilled: res => {
console.log(res)
            this.eMessage = '';
            this.sMessage = 'Login Successfull';
            alert(this.sMessage);
            this.router.navigate( commands: ['/home']);
        }, onrejected: err => {
            console.log(err);
            this.eMessage = err.message;
```

LoginComponent

Register.html

```html
<div class="form-content">
  <div class="row">
    <div class="col-md-6">
      <div class="form-group">
        <input type="text" class="form-control" placeholder="Your Full Name" [(ngModel)]="fullname" name="fullname" />
      </div>
      <div class="form-group">
        <input type="text" class="form-control" placeholder="Email_id"  [(ngModel)]="email" name="email"/>
      </div>
    </div>
    <div class="col-md-6">
      <div class="form-group">
        <input type="password" class="form-control" placeholder="Your Password" [(ngModel)]="password" name="password" /
      </div>
    </div>
  </div>
  <span>
  <button type="button" class="btnSubmit" (click)="register()">Register</button>
    <a routerLink="/login" style="..."><i>login</i></a>
  </span>
</div>
div>
v>
```

Register.ts

```
public class;
public password;
public errorMessage;
public successMessage;
constructor(public auth: FireauthService, public router: Router) { }

ngOnInit() {
}

register() {
  this.auth.fireRegister(this.email, this.password).then( onfulfilled: res => {
    console.log(res)
    this.errorMessage = '';
    this.successMessage = 'Accout got Created';
    alert(this.successMessage);
    this.router.navigate( commands: ['/login']);
  }, onrejected: err => {
    console.log(err);
    this.errorMessage = err.message;
    this.successMessage = '';
    alert(this.errorMessage);
  });
}
}
```

fireauthservice

```
1  import { Injectable } from '@angular/core';
2  import { AngularFireAuth } from 'angularfire2/auth';
3  import * as firebase from 'firebase/app';
4
5  @Injectable({
6    providedIn: 'root'
7  })
8  export class FireauthService {
9    constructor(public auth: AngularFireAuth) {
10   }
11   fireRegister(v1, v2) {
12     return new Promise<any>( executor: (resolve, reject) => {
13       firebase.auth().createUserWithEmailAndPassword(v1, v2).then( onfulfilled: res => {
14         resolve(res);
15       }, onrejected: err => reject(err));
16     });
17   }
18
19
20   firelogin(v1, v2) {
21     return new Promise<any>( executor: (resolve, reject) => {
22       firebase.auth().signInWithEmailAndPassword(v1, v2).then( onfulfilled: res => {
23         resolve(res);
24       }, onrejected: err => reject(err));
25     });
26   }
27   firesignout() {
28     return new Promise<any>( executor: (resolve, reject) => {
```

appModule

```
12      import {FormsModule} from '@angular/forms';
13      import {FireauthService} from './fireauth.service';
14      import { SidebarDirective } from './sidebar.directive';
15
16      @NgModule({
17        declarations: [
18          AppComponent,
19          LoginComponent,
20          RegisterComponent,
21          HomeComponent,
22          SidebarDirective
23        ],
24        imports: [
25          BrowserModule,
26          FormsModule,
27          AngularFireModule.initializeApp(environment.firebase),
28          AngularFirestoreModule,
29          AngularFireAuthModule,
30          AppRoutingModule],
31        providers: [],
32        bootstrap: [AppComponent]
33      })
34      export class AppModule { }
35
```