Simple Ionic Application

Team: ID: 6-1

Sindhusha Tiyyagura Class ID:24

Pradeepika Kolluru Class ID:12

Introduction:

Creating a basic android application using Ionic Framework (login, signup pages with a mashup of 2 web service pages)

Main Objective:

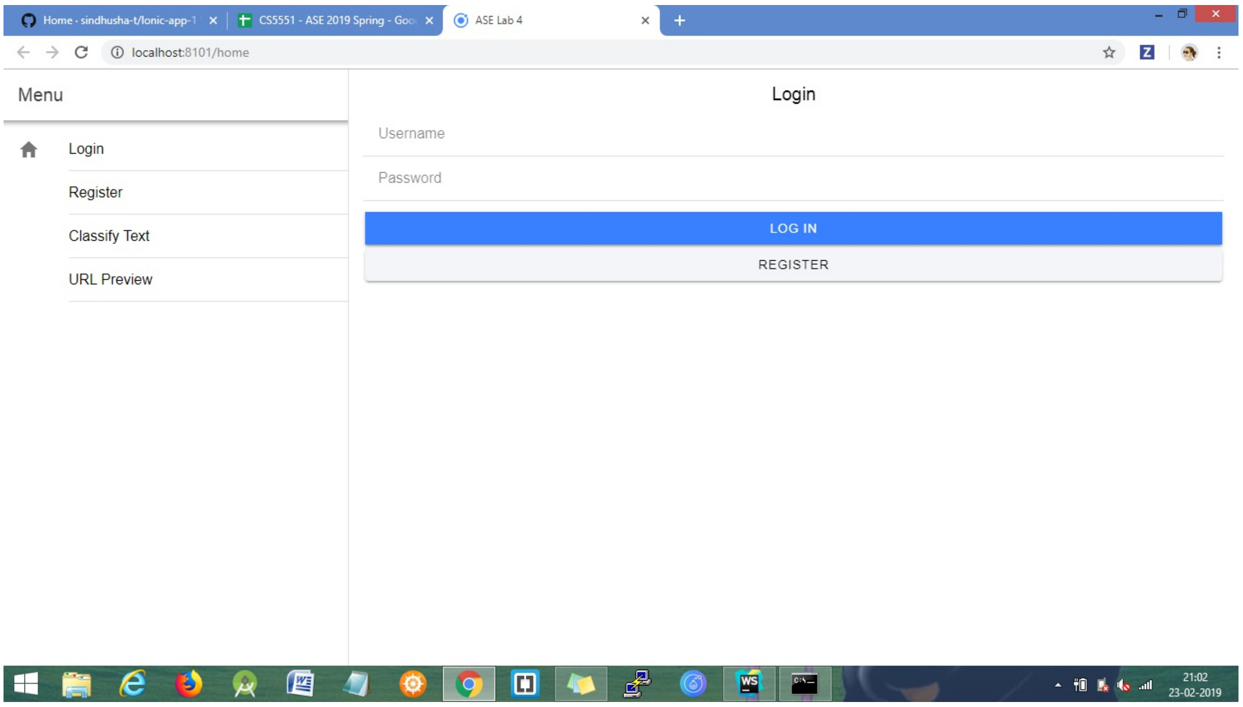
Task-1: Creating basic Log In and Sign Up pages using ionic tags and local storage functionalities
Task-2: Creating a mashup of 2 web service APIs related to the knowledge services/AI/machine Learning
Task-3: Creating Unit Testing environment using Jasmine and Karma, and Writing at least 3 unit test cases for the ionic application.

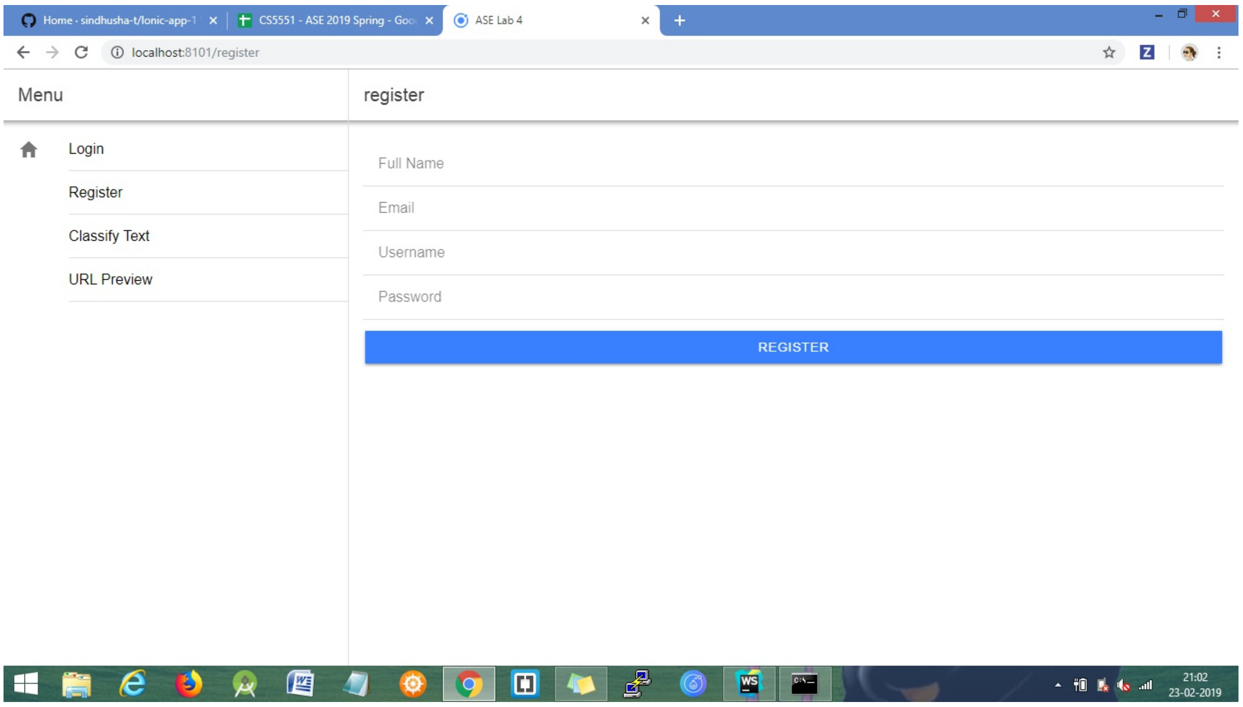
Design and Implementation for each of the above tasks:

Task-1: Creating basic Log In and Sign

Up pages using ionic tags and local storage functionalities:

1. Created Ionic application using command in command prompt: `ionic start ionic-app sidemenu`
2. Opened the Ionic project in Webstorms.
3. Created 2 new pages for Log In and Sign Up using command:
   `ionic g page login`
   `ionic g page register`
4. Added the two pages in the app.routing.ts file for displaying the menu with login and register pages.
5. Added Basic html page in log in and sign up pages using ionic html tags
6. Created AuthService provider using command: `ionic g service AuthService`
7. Providing 2 functions in the AuthService with the local storage functionality to set and get the login details.
8. Respective AuthService functions are called from the login and signup pages for registering the user details and checking the credentials of the user.
9. After registering user details successfully, page will be redirected to login page using Router Module.
10. Running the ionic appication using command `ionic serve`





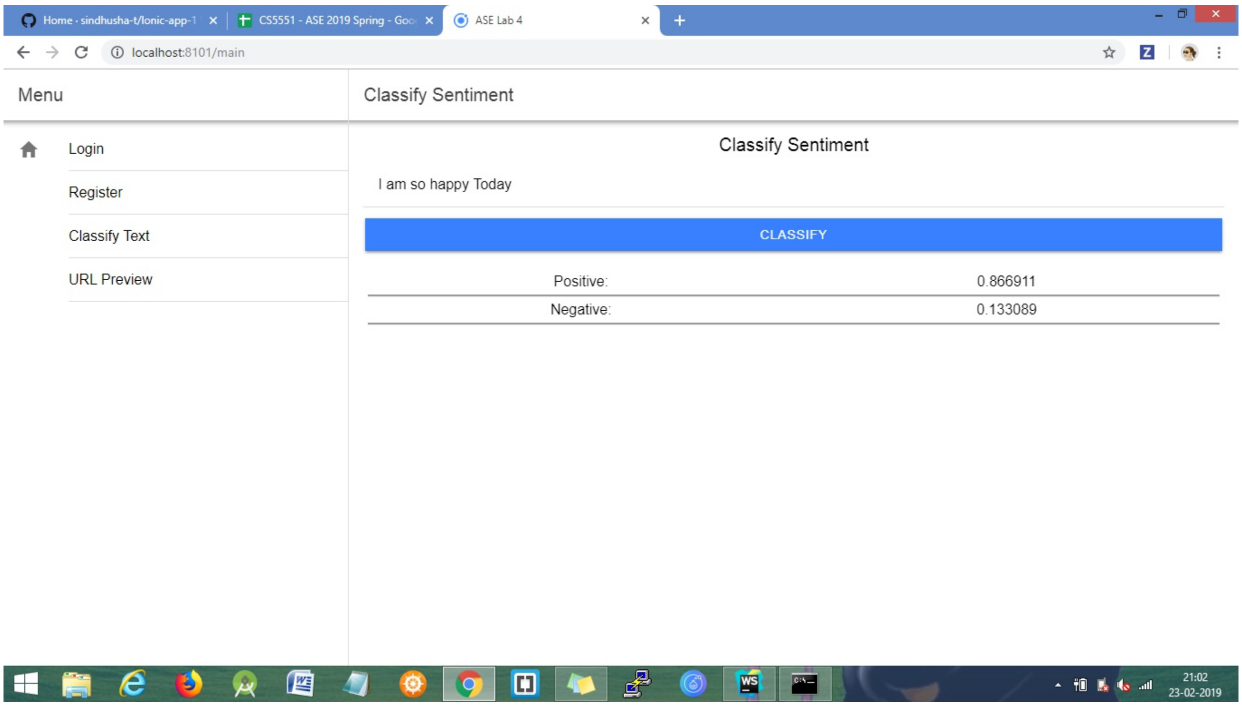
Task-2: Creating a mashup of 2 web service APIs related to the knowledge services/AI/machine Learning:

1. Selected the following web services for the ionic app:

   Classify API: https://www.uclassify.com/

URL Preview API: https://www.microsoft.com/cognitive-services/en-us/academic-knowledge-api

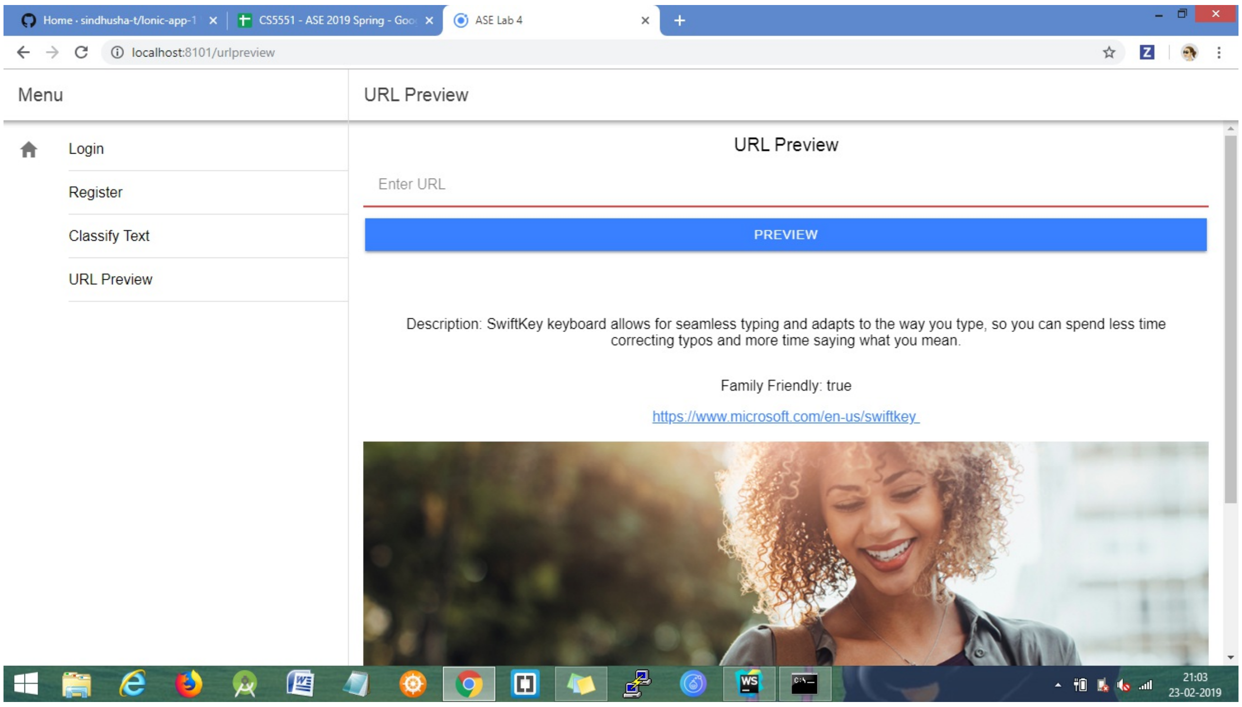
2. Created an account in the above mentioned web services and generated keys.
3. Created 2 new pages for the following web services using command: `ionic g page classifyText` `ionic g page urlPreview`
4. Written basic html page templates for both the web service pages.
5. For Classify API page: When user enters a sentence, Http request is sent to the classify API url and gives the scale of positive and negative impact of the sentence in values ranging from (0 to 1)
6. Displaying the above contents in a table using ionic html tags
7. For URL Preview API page: When user enters the URL in the input, HTTP get request is sent to the web service URL using the subscription key specified in the HttpHeader and gives the response back with the following details:
    - Short description about URL
    - Whether the URL is family Friendly or not.
    - URL
    - Image related to the URL
8. Both pages were added in the app.routing.ts file to display in the side menu items list.



Classify Sentiment

I am so happy Today

Positive:	0.866911
Negative:	0.133089

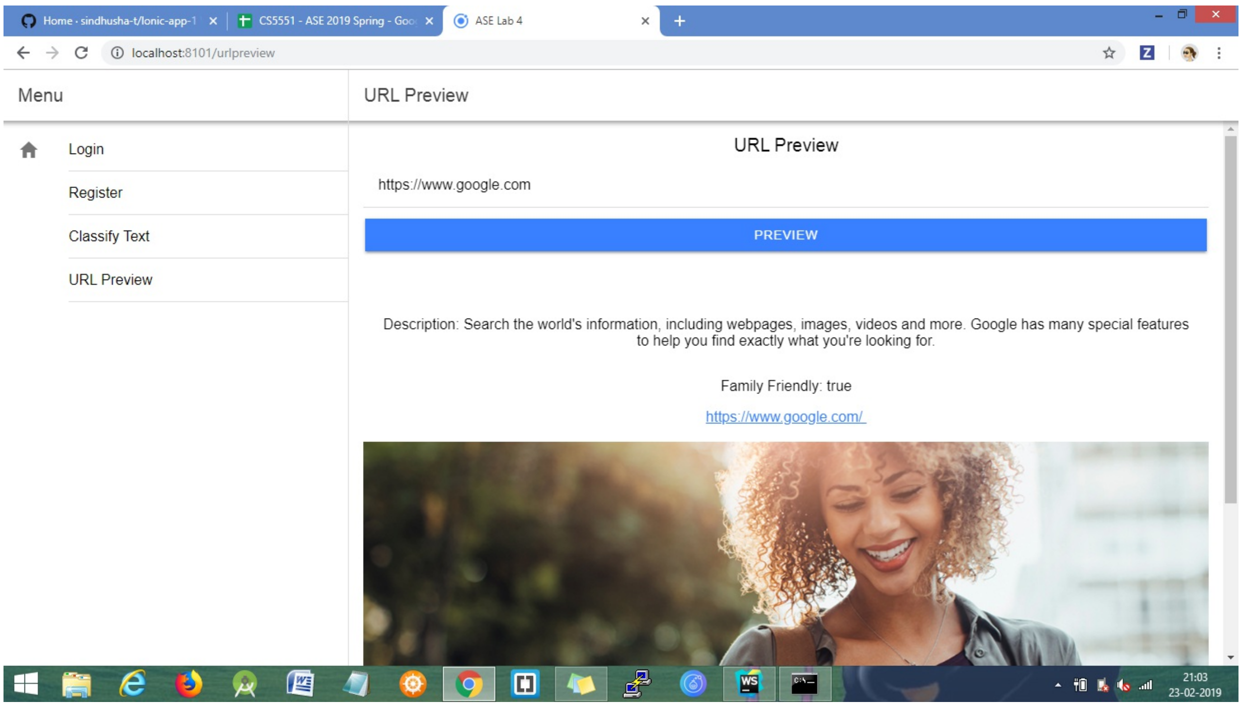


URL Preview

Description: SwiftKey keyboard allows for seamless typing and adapts to the way you type, so you can spend less time correcting typos and more time saying what you mean.

Family Friendly: true

https://www.microsoft.com/en-us/swiftkey


Home · sindhusha-t/Ionic-app-1
CS5551 - ASE 2019 Spring - Goo
ASE Lab 4
localhost:8101/urlpreview
Menu
URL Preview
Login
Register
Classify Text
URL Preview
PREVIEW
21:03
23-02-2019

URL Preview

https://www.google.com

Description: Search the world's information, including webpages, images, videos and more. Google has many special features to help you find exactly what you're looking for.

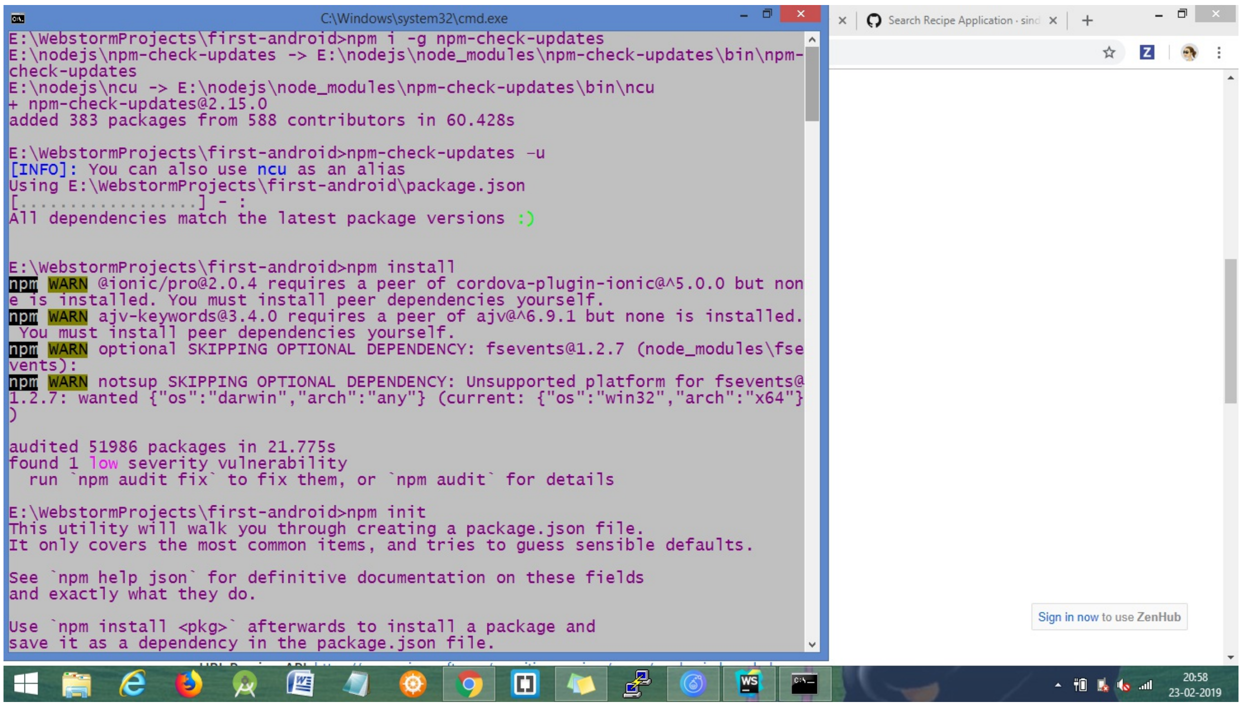
Family Friendly: true

https://www.google.com/

Task-3: Creating Unit Testing environment using Jasmine and Karma, and Writing at least 3 unit test cases for the ionic application.

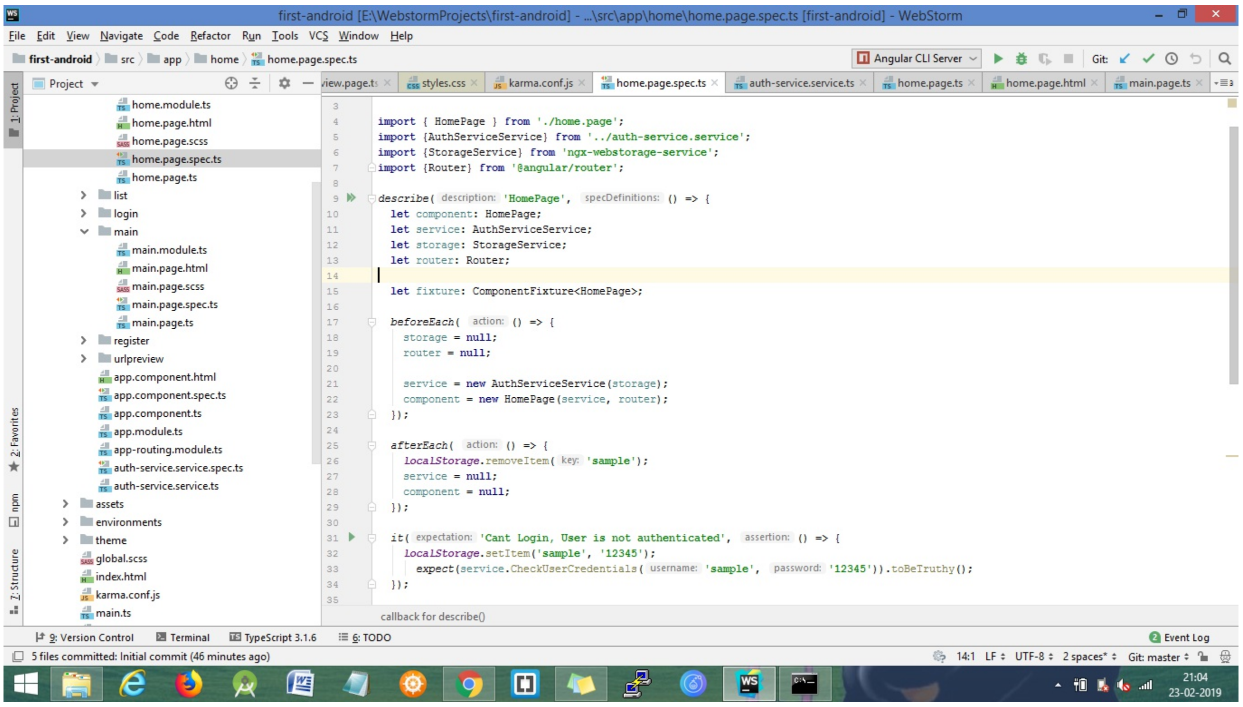
1. Karma and Jasmine testing tools are used for testing angularjs applications.
2. Created Karma default config file using command: `karma init`
3. We have written unit testing for login page.
4. We have set localstorage with user details and verified credentials function.
5. We have written unit testing for registration page by executing saveUserDetails function and expecting to have the values in the localstorage function.
6. All the values set in the local storage will be removed after the

unit testing is done.



```
E:\WebstormProjects\first-android>npm i -g npm-check-updates
E:\nodejs\npm-check-updates -> E:\nodejs\node_modules\npm-check-updates\bin\npm-
check-updates
E:\nodejs\ncu -> E:\nodejs\node_modules\npm-check-updates\bin\ncu
+ npm-check-updates@2.15.0
added 383 packages from 588 contributors in 60.428s

E:\WebstormProjects\first-android>npm-check-updates -u
[INFO]: You can also use ncu as an alias
Using E:\WebstormProjects\first-android\package.json
[..................] - :
All dependencies match the latest package versions :)


E:\WebstormProjects\first-android>npm install
npm WARN @ionic/pro@2.0.4 requires a peer of cordova-plugin-ionic@^5.0.0 but non
e is installed. You must install peer dependencies yourself.
npm WARN ajv-keywords@3.4.0 requires a peer of ajv@^6.9.1 but none is installed.
 You must install peer dependencies yourself.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.7 (node_modules\fse
vents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@
1.2.7: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"}
)

audited 51986 packages in 21.775s
found 1 low severity vulnerability
  run `npm audit fix` to fix them, or `npm audit` for details

E:\WebstormProjects\first-android>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.
```

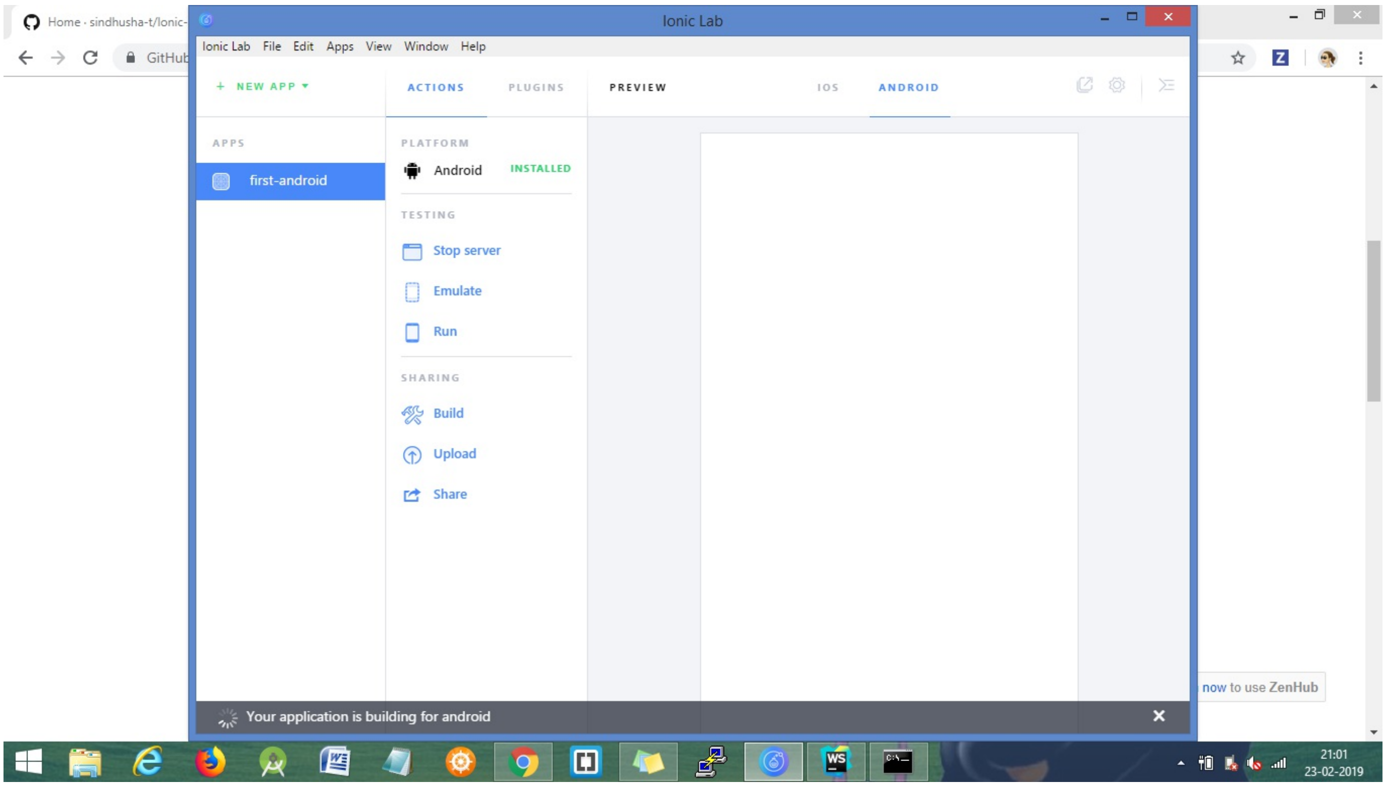
C:\Windows\system32\cmd.exe
Search Recipe Application · sinc
Sign in now to use ZenHub
20:58
23-02-2019



first-android [E:\WebstormProjects\first-android] - ...\src\app\home\home.page.spec.ts [first-android] - WebStorm

```
import { HomePage } from './home.page';
import {AuthServiceService} from '../auth-service.service';
import {StorageService} from 'ngx-webstorage-service';
import {Router} from '@angular/router';

describe( description: 'HomePage',  specDefinitions: () => {
  let component: HomePage;
  let service: AuthServiceService;
  let storage: StorageService;
  let router: Router;

  let fixture: ComponentFixture<HomePage>;

  beforeEach( action: () => {
    storage = null;
    router = null;

    service = new AuthServiceService(storage);
    component = new HomePage(service, router);
  });

  afterEach( action: () => {
    localStorage.removeItem( key: 'sample');
    service = null;
    component = null;
  });

  it( expectation: 'Cant Login, User is not authenticated',  assertion: () => {
    localStorage.setItem('sample', '12345');
      expect(service.CheckUserCredentials( username: 'sample',  password: '12345')).toBeTruthy();
  });
```

Deployed the basic Ionic application in Ionic Lab.



Conclusion:

We have learnt how to write an android applications using Ionic. It is almost similar to developing web applications just that html pages differ by the tags.

We have learnt how to write unit testing using karma and Jasmine.