

Angular POC Project

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology
in

Computer Science and Engineering/Information Technology

By

Aviral Saini (151452)

Kamal Jeet Singh(151262)

Under the supervision of

Jatin Kapil

(Technical Lead, Project Development Department, Click Labs)

To



Department of Computer Science & Engineering and Information Technology

CERTIFICATE

**Jaypee University of Information Technology Wagnaghat, Solan-173234,
Himachal Pradesh**

Candidate's Declaration

I hereby declare that the work presented in this report entitled “**Angular POC Project**” submitted in the department of Computer Science & Engineering and Information Technology, **Jaypee University of Information Technology, Wagnaghat** is an authentic record of my own work carried out under the supervision of **Mr. Jatin Kapil** (Technical Lead, Project Development Department, Click Labs). The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Aviral Saini (151452)

Kamal Jeet Singh (151262)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Mr. Jatin Kapil

Technical Lead-Project Development Department

Click Labs, Chandigarh

Date: May,2019

ACKNOWLEDGEMENT

I would like to extend my deep gratitude to my project mentor Mr. Jatin Kapil Tech.Lead at Click Labs, Chandigarh who gave me a golden opportunity to get a hands on experience in Frontend Development as as Software Engineer, which helped me to develop many new skills in technologies as well as enhanced our self-development skills that will help build my career ahead. We get to know about working in team and had a great IT industry exposure.

I would also like to extend my sincere thanks to all the faculty and staff of the Department of Computer Science and Information Technology of Jaypee University of information and Technology for all the support and motivation provided by them.

LIST OF FIGURES

FIGURES	PAGE	REFRECE
Figure 1.1	8	https://jungleworks.com
Figure 1.2	9	https://jungleworks.com/
Figure 2.1	14	Project Screenshot
Figure 2.2	15	Project Screenshot
Figure 2.3	16	Project Screenshot
Figure 2.4	17	Project Screenshot
Figure 2.5	19	Project Screenshot
Figure 3.1	22	https://nativescript.org/core-concept/technical-overview
Figure 3.2	25	Project Screenshot
Figure 3.3	26	Project Screenshot
Figure 3.4	27	Project Screenshot
Figure 3.5	27	Project Screenshot

TABLE OF CONTENTS

Certificate	2
Acknowledgment	3
List of Figures	4
Abstract	7
1. Chapter-1 About the Company	
1.1 Jgunoo	8
1.2 Jungle Works.....	9
1.3 Inter Front-End Development.....	10
1.4 Responsibilities.....	10
2. Chapter-2 Angular POC	
2.1 Introduction.....	11
2.2 Objective.....	12
2.3 Tools & Frameworks	
2.3.1 HTML.....	12
2.3.2 CSS.....	13
2.3.3 Angular 2+.....	13
2.4 Application Designing	
2.4.1 Registration Page.....	14
2.4.2 Login Page.....	15
2.4.3 Profile Page.....	16

2.4.4 Booking Page.....	17
2.4.5 FAQ's Page.....	18
2.4.6 Auth Guards.....	19
2.4.7 Services.....	20
2.5 Commands.....	21
3. Chapter-3 Native Script	
3.1 Introduction.....	22
3.2 Core Modules.....	23
3.3 Layouts.....	23
3.4 Objectives.....	24
3.5 Application Designing	
3.5.1 Login Page.....	25
3.5.2 Dashboard.....	26
3.5.3 Add Team.....	27
3.5.4 Update Team.....	28
3.6 Commands	28
3.7 Code Splitting.....	29
4 Chapter-4 Result and Performance Analysis	30
5 Conclusion	32
References	

ABSTRACT

A front-end web developer is probably what most people think of as a “web developer”. A front-end web developer is responsible for implementing visual elements that users see and interact with in a web application. They are usually supported by back-end web developers, who are responsible for server-side application logic and integration of the work front-end developers do. The Logics for UI are implemented by Front End Developer with the help of different frameworks present in the market.

As for now NativeScript is getting the market to its corner but still ANGULAR plays a Vibrant Role.

CHAPTER-1

ABOUT THE COMPANY

Click Labs is Chandigarh based company that provide high grade generation answer with a focal point on mobility. Thae company grants mobility solutions to marketers and organizations trying to provide an Omni channel to their customers with a goal to engineer & maximize mobile ROI. company's commercial enterprise approach is to increase on call for custom enterprise solutions on pinnacle of proprietary core IP merchandise for mobile paintings force control.

Click Labs has two major subsidiaries named JUGNOO and JUNGLEWORKS

1.1 Jugnoo

Jugnoo, a wonder that was begun as an auto-aggregator, is presently one stage answer for every nearby need. The beginning is to provide comfortable & bother free life by conveying the finest of each metropolitan to its occupants. **Started in November 2014** by pack of IITians, Jugnoo pushed toward getting to be as Country's



primary auto rickshaw aggregator with the impartial of constructing each day drive simple and solid. In the wake of convey the on-interest insurgency to day by day drive, we extended our vision and dream of contacting regular day to day existences in a lot greater way. With 10lac exchanges every month crosswise over 45+ urban areas, however started as a little organization in 2014 is presently pear forwarding the development of nearby Indian commercial centre into advanced. Jugnoo now stretches a extensive possibility of neighbourhood profits on interest auto rides, basic supplies, wherewithal transport, dinners, coordination and even a range based adjoining informal community. It gives on-request transportation and coordination benefits crosswise over 38 urban areas in India through their portable tender that can be moved in Android, iOS and Windows based telephones.

1.2 Jungle Works

Wilderness Works is a suite of upsetting items that centres around each progression of an on-request business venture. Our central goal is to make the best client involvement with each progression. It has different item that has truly give new client experience to both end client and just as the business people. It gives different items that has altered the creation and has acquired a major modification the world itself.

The different ventures of the JungleWorks are:



BI Analytics Dashboard

Easily prepare, analyze & explore big data from multiple sources. Give power to every user to easily discover meaningful business insights.



Sales CRM

AI-powered CRM for smart sales teams. Get insights about your business, predict customer behavior, and build a scalable sales organization.

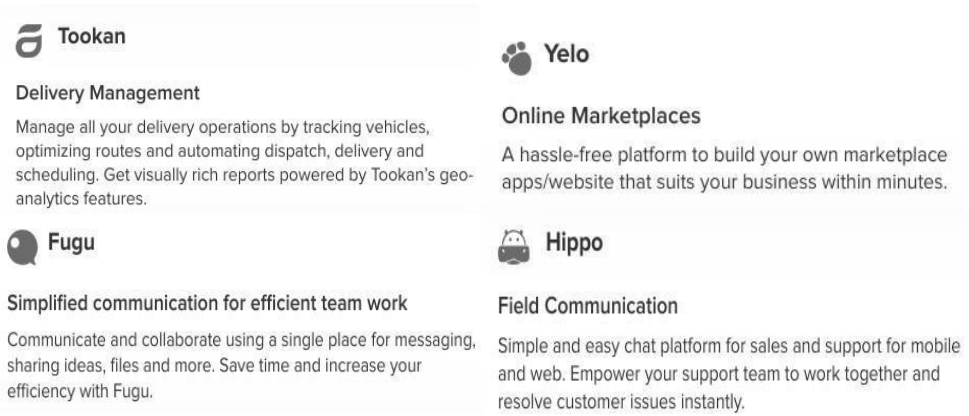


Figure 1.2 Jungleworks Products

1.3 Intern-Front End Developer

A front-end web designer is likely what a great many folks consider as a "web engineer". A front-end web designer is in charge of actualizing pictorial elements that clients can view & communicate with in a web application. They are normally bolstered by back-end web engineers, who are in charge of server-side application rationale and joining of the work front-end designers do. The Logics for UI are executed by Front End Developer with the assistance of various systems present in the market. Concerning now NativeScript is getting the market to its corner yet at the same time ANGULAR assumes a Vibrant Role

1.4 Responsibilities

- Develop new client confronting highlight.
- Build reusable code and libraries for some time later
- Ensure the specialized attainability if UI/UX plan.
- Optimize application for greatest speed and versatility
- Assure that all client input is approved before submitting to back end
- Collaborate with other colleague and partners

I chipped away at Angular-7 CLI and aided in Companies Projects, the name for such Projects can't be uncovered according to organizations Norms. Front End Development is as much significant as Backend Developer as UI Implementation is the primary thing a User gain admittance to.

CHAPTER: 2

ANGULAR POC

1. INTRODUCTION

AngularJS used to be the "prodigy" among JavaScript structures, as it was at first presented by Google partnership in 2012. It was worked in light of the Model-View-Controller idea, however creators of the structure regularly called it "Model-View-*" or even "Model-View-Whatever".

The structure, written in unadulterated JavaScript, was expected to decouple an application's rationale from DOM control, and went for dynamic page refreshes. In any case, it wasn't exceptionally meddlesome: you could have just a piece of the page constrained by AngularJS. This structure presented numerous ground-breaking highlights enabling the designer to make rich, single-page applications effectively.

AngularJS ended up famous in all respects rapidly and got a great deal of footing. All things considered, its maintainers chose to make another stride further and continued to build up another rendition which was at first named Angular 2 (later, basically Angular without the "JS" part). It's no fortuitous event the structure got another name: really, it was completely re-composed and overhauled, while numerous ideas were re-examined.

It's JavaScript, yet better. Precise is worked with TypeScript, which thusly depends on JS ES6. You don't have to become familiar with an absolutely new dialect, yet despite everything you get highlights like static composing, interfaces, classes, namespaces, decorators and so on.

The AngularJS structure includes the Hypertext Mark-up Language also known as HTML for pages visible to the user. AngularJS also include Java

Script variables to bind the input supplied by the user and formulate the output.

2.2 OJECTIVE

Precise POC Project was implemented using Angular Framework with rendition 7. The Project was supplied by the tutor to highlight the structure of the framework so that new interns can be adjusted on different live Projects of the organization. The main goal was to get us familiar with logics, for example,

- User Authentication
- Auth Guards
- Client Services
- Controllers.
- Database
- Functionality Module

2.3 TOOLS & FRAME WORKS

2.3.1 HTML:

Hypertext Markup Language (HTML) is the standard markup language for making website pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it frames a set of three of foundation advancements for the World Wide Web.

Internet browsers get HTML archives from a web server or from neighbourhood stockpiling and render the records into sight and sound website pages. HTML depicts the structure of a website page semantically and initially included signs for the presence of the record.

2.3.2 CSS:

Cascading Style Sheets (CSS) is a template language utilized for adding elements of graphic appeal to the website that makes it more user friendly by improving the interface for user interaction. CSS can be integrated into an HTML page in three ways

- External Style Sheet
- Inline Style Sheet
- Embedded CSS

2.3.3 ANGULAR (web framework):

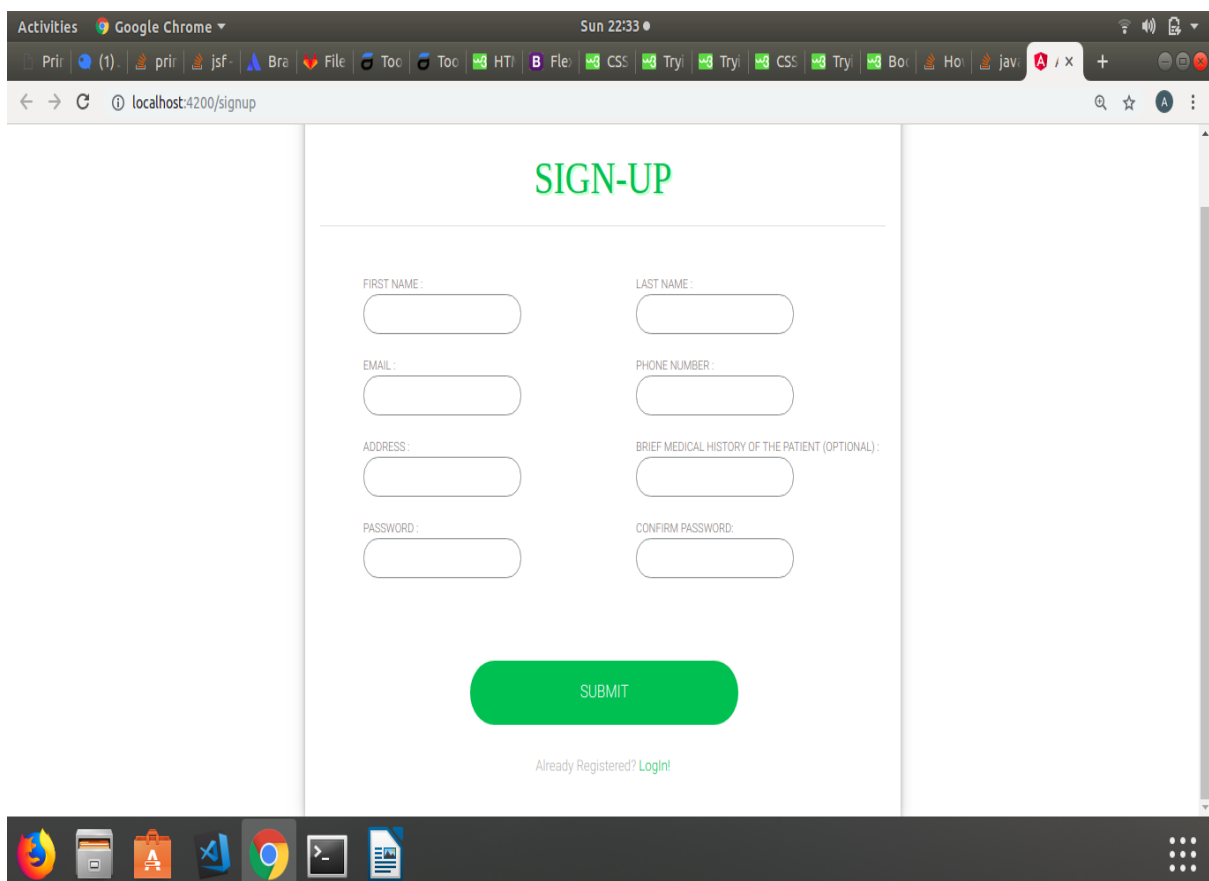
Angular (ordinarily alluded to as "Angular 2+" or "Angular v2 or more") is a TypeScript-based open-source web application system driven by the Angular Team at Google and by a network of people and companies. Angular is a finished rework from a similar group that fabricated AngularJS. Due to its popularity, it has been in use extensively as a client-side framework. Google announced AngularJS in 2012. AngularJS is an adaptation of the mainstream Model-View-Controller design (Model-View-*).

The AngularJS system, was based over JavaScript with the mean to decouple the business rationale of an application from the low dimension DOM control and make dynamic sites.

2.4 APPLICATION DESIGN

2.4.1 REGISTRATION PAGE:

The Registration Page incorporated into project so that the new client can provide the data and store it in the Database with the help of a Post Service. The Post API assembles entire of the data and store it in the database which was made by the use of JSON-SERVER. This Page is a Part of Public Module.



The image shows a screenshot of a web browser displaying a registration form. The browser's address bar shows the URL 'localhost:4200/signup'. The form is titled 'SIGN-UP' in green text. It contains several input fields: 'FIRST NAME', 'LAST NAME', 'EMAIL', 'PHONE NUMBER', 'ADDRESS', 'BRIEF MEDICAL HISTORY OF THE PATIENT (OPTIONAL)', 'PASSWORD', and 'CONFIRM PASSWORD'. A green 'SUBMIT' button is located at the bottom of the form. Below the button, there is a link that says 'Already Registered? Login!'. The browser's taskbar at the bottom shows various application icons.

Fig 2.1 Registration Form

If anyone is registered, they can choose to be re-directed to the login page.

This Page incorporates label elements and distinctive input fields. The input of Password and Password recheck are associated with Custom Validator. The register button will appear only once all the field are valid, i.e. for each input field obeys the set validation.

2.4.2 LOGIN PAGE:

The Login Page incorporated into application uses HTML's GET Method to fetch the details from the API and then validates them in backend to verify if credentials are correct. In the credentials are correct the system will fetch all data associated with the concerned credentials. This page is included under public module since it is accessible publicly.

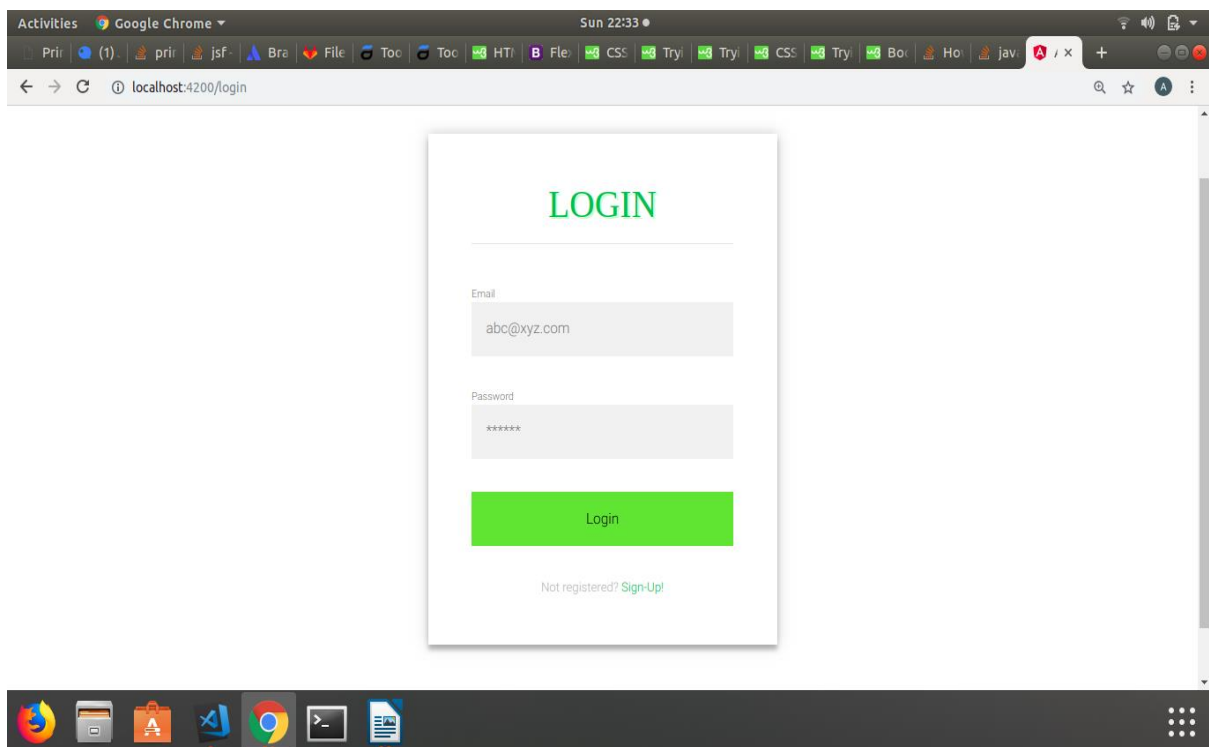


Fig 2.2 Login Page.

The Login Button Will be incapacitated till it get the required certifications. In the event that and New User get straightforwardly to Login Page the connection to enlist, is likewise given in the format. In the event that any client isn't enrolled and attempts to login The Alert concocts message "You are not enlisted".

2.4.3 PROFILE PAGE:

Once the user has successfully logged in he will be redirected to his profile view that displays all his personal information. This information is editable as per user's requirement. When the user makes an update to his profile the patch is escalated to carry out the change in the database.

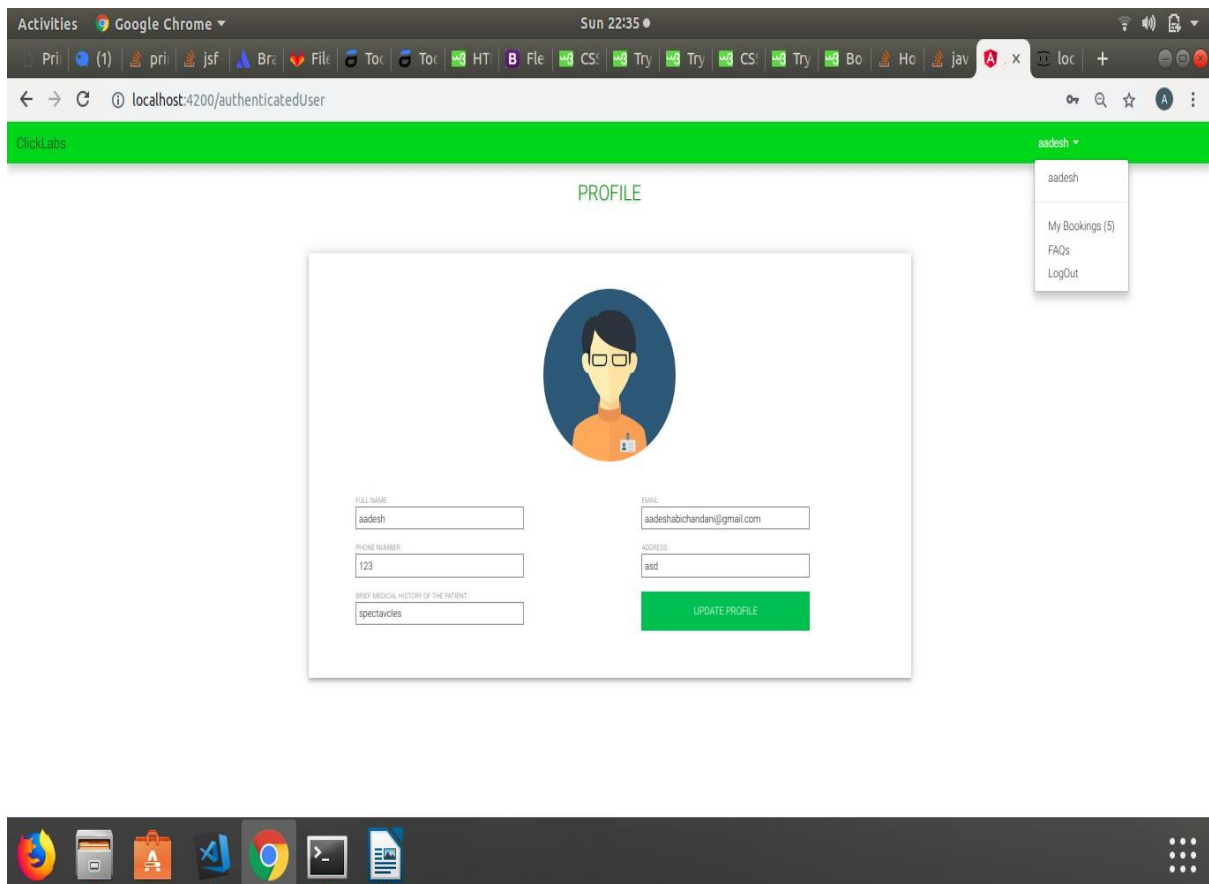


Fig 2.3 Profile Page

The profile also includes a display for the name of the currently logged in user. This is made functional by the use of two-way binding.

The Navigation-bar includes a dropdown list that includes options namely My Bookings, FAQ, Log out. Each of this option is backed by a link that re-directs a user to an action on page of concern. Since all of these options should be accessible upon login therefore, to ensure their restricted access these have been included in the private module. The detailed working of options My Booking and FAQ has been described below

2.4.4 BOOKING PAGE:

This page allows the user to access all the booking with his credentials these include both past as well as up-coming bookings. To fetch these details the modules involves the use of Get Service to fetch the Json Object which contains all the Booking information.

Bookings Page comprises of two Hyperlinks:

1. Past
2. Upcoming

Since Past and Upcoming bookings are distinct of each other thus both them are associated with different views.

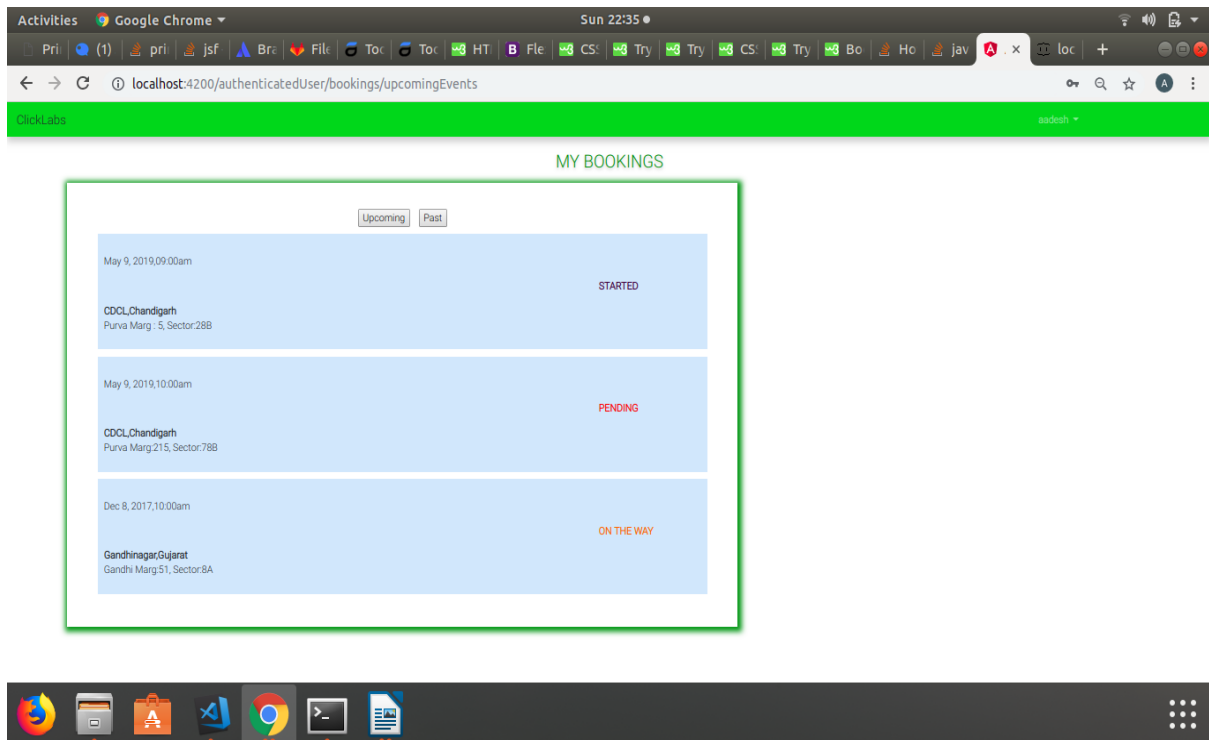


Fig 2.4 Booking Page

2.4.5 FAQ's PAGE :

FAQ Page depends on Accordion Structure. The Accordion columns clicked once hit the Get Service get the information from the database. The Accordion include gains admittance to one line at any given moment.

Inquiries incorporated into with various inquiries posed often.

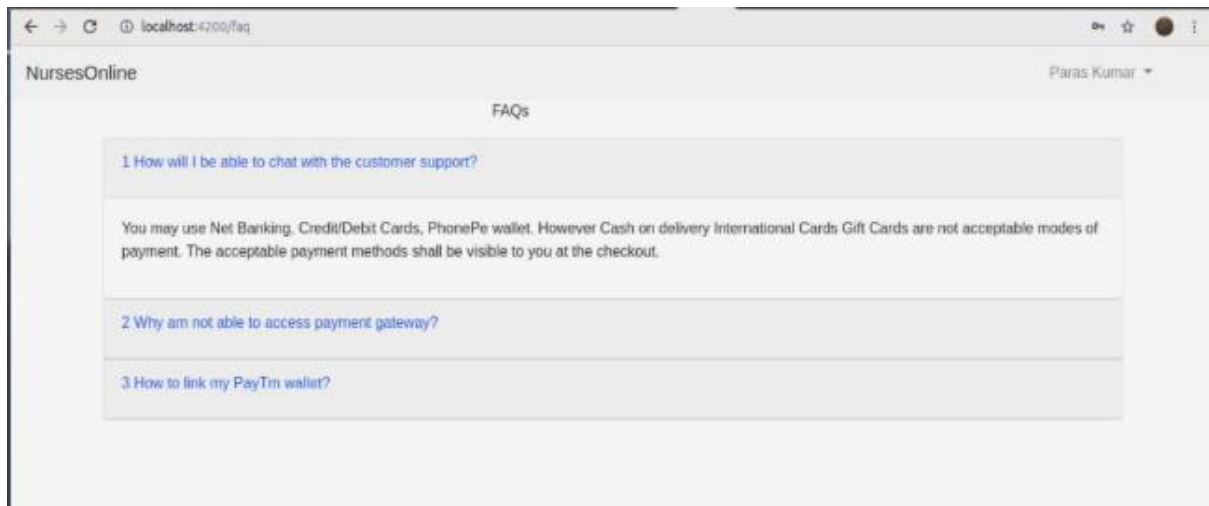


Fig 2.5 FAQ's

2.4.6 AUTH GAURDS :

Rakish course protects are interfaces which can tell the switch whether it ought to enable route to a mentioned course. They settle on this choice by searching for a genuine or false return an incentive from a class which executes the given gatekeeper interface.

CAN-ACTIVATE auth watch is utilized in the application with a condition that if a client is signed in, navbar part and profile segment get render on the screen. In the event that anybody endeavors to get into the Application without logging, the individual ought not have the option to enter in application. Auth Guards spares the application from the unauthentic client. Many Different Auth Guards are there in angular for different authentication purposes:

- CanActivate
- CanActivateChild
- CanDeactivate
- CanLoad
- Resolve

2.4.7 SERVICES :

Service is a general classification enveloping any esteem, capacity, or highlight that an application needs. A service is commonly a class with a restricted, well-characterized reason. It ought to accomplish something explicit and do it well. Angular recognizes segments from services to expand measured quality and reusability. By isolating a part's view-related usefulness from different sorts of preparing, you can make your segment classes lean and productive.

In a perfect world, a part's main responsibility is to empower the client experience and nothing more. A part should exhibit properties and strategies for information official, so as to intervene between the view (rendered by the layout) and the application rationale (which frequently incorporates some idea of a model).

A segment can designate certain assignments to services, for example, bringing information from the server, approving client information, or logging straightforwardly to the comfort. By characterizing such handling assignments in an injectable service class, you make those undertakings accessible to any segment. You can likewise make your application progressively versatile by infusing various suppliers of a similar sort of service, as suitable in various conditions.

Precise doesn't uphold these standards. Precise helps you pursue these standards by making it simple to consider your application rationale services and make those services accessible to segments through reliance infusion.

2.5 COMMANDS :

There are so different commands in Microsoft Visual Studio Code for Angular CLI in order to generating different modules of project like feature-module , components , services , authguards , and even to run an application we have to run a particular command so that it can run on web-browser. These commands also inject the dependency itself in application module whenever we run “generate” command.

The list command used in this Application with their purpose are given below:

- Command for generating a Feature Module:
ng generate module module_name
- Command for creating Component:
ng generate component component_name
- Command for creating Services:
ng generate service service_name
- Command to run an app:
ng serve –open

CHAPTER : 3

NATIVE SCRIPT

1. INTRODUCTION

NativeScript is an open-source structure used to create applications that runs successfully on both the Android and iOS. It is also use for developing web application simultaneously with mobile application. It was initially visualised and formulated step by step. This Script's applications are written in JavaScript, or languages equivalent to JavaScript, for example TypeScript. This Script is compatible with both Angular as well as Vue JavaScript systems. The Portable mobile applications delevoped using NativeScript result in entirely local applications, similar to the APIs developed using Xcode or Android Studio. It allows programer to use additional programming libraries from portalssuch as npm.js, Android Arsenal, CocoaPods or others elimanating the wrappers. The NativeScript includes modules like - CLI, Runtimes, Plugins Core Modules.

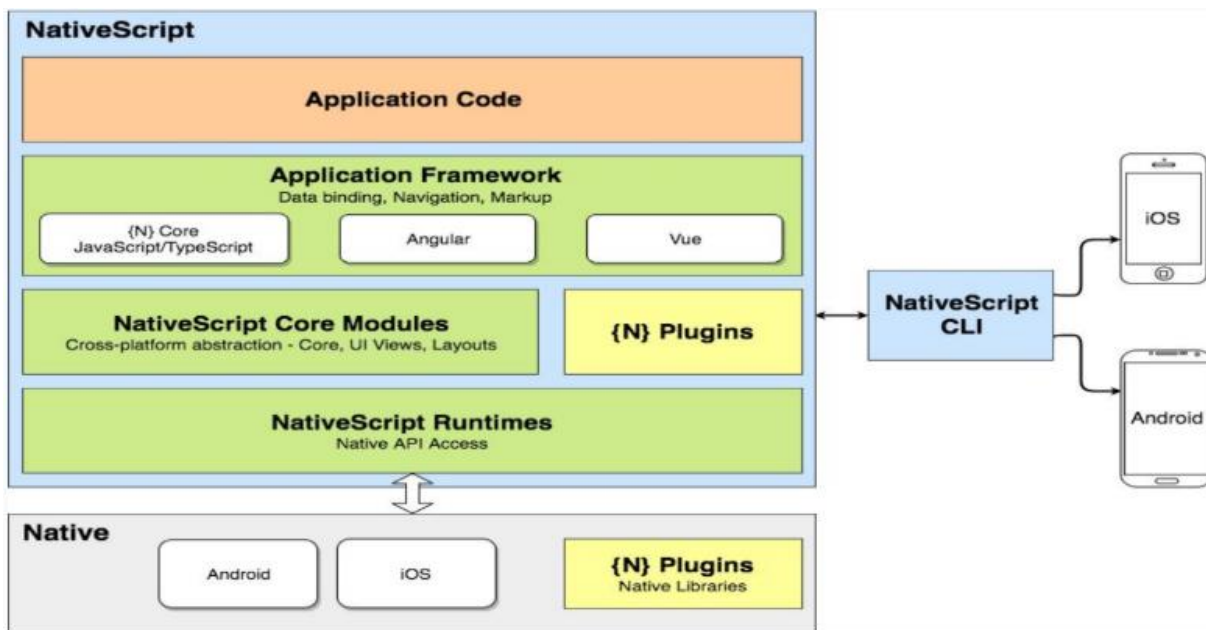


Figure 3.1 Nativescript Architecture

3.2 CORE MODULES

application: Gives the application deliberation every related strategy.

console: Lets you print messages to the gadget comfort.

application-settings: Gives you a chance to spare and reestablish any data identified with your application.

http: Gives you a chance to send web demands and get the reactions.

image-source: Gives the ImageSource class which typifies the basic reflection behind a stage explicit item that is utilized as a hotspot for pictures (ordinarily a Bitmap).

timer: Lets you make, begin, stop and respond to clocks.

trace: Gives you a chance to follow and print explicit data dependent on classes.

ui/image-cache: Gives the Cache class which handles picture download demands and stores the as of now downloaded pictures.

connectivity: Gives you a chance to check the sort of Internet association and screen its state changes

3.3 LAYOUTS

ui/layouts/stack-layout: Gives the StackLayout class which gives you a chance to orchestrate the offspring of the design in a solitary line.

ui/layouts/grid-layout: Gives the GridLayout class which gives you a chance to mastermind the offspring of the format in an adaptable matrix region with segments and columns.

ui/layouts/absolute-layout: Gives the AbsoluteLayout class which gives you a chance to mastermind the offspring of the format at discretionary positions or attract them numerous layers.

ui/layouts/wrap-layout: Gives the WrapLayout class which gives you a chance to mastermind the offspring of the format at consecutive positions from left to right and after that wrap the lines of youngsters through and through.

ui/layouts/dock-layout: Gives the DockLayout class which gives you a chance to orchestrate the offspring of the design at top, base, left and right of the format.

3.4 OBJECTIVE

The basic objective of this project is similar to one highlighted before in the report earlier, i.e. to build an application on angular framework version 7. In this project the application was build to track the sales of each and every team in the sales department. After successful login the portal, this application presents a list of the teams along with their total sale of the individual team as well the company consideration. It is to be noticed that the admin has the additional rights to manage the teams. The features included are add , update and delete team details.

3.5 APPLICATION DESIGN

The Shared Application in discussion includes different modules like Login, Personal/Admin Dashboard, in addition to feature that includes features for team management

3.5.1 LOGIN PAGE:

The login page includes two text fields one for username and one for password that are used to validate user's credentials to re-direct him to designated dashboard. Given below is a screenshot that establishes the view presented to user for login

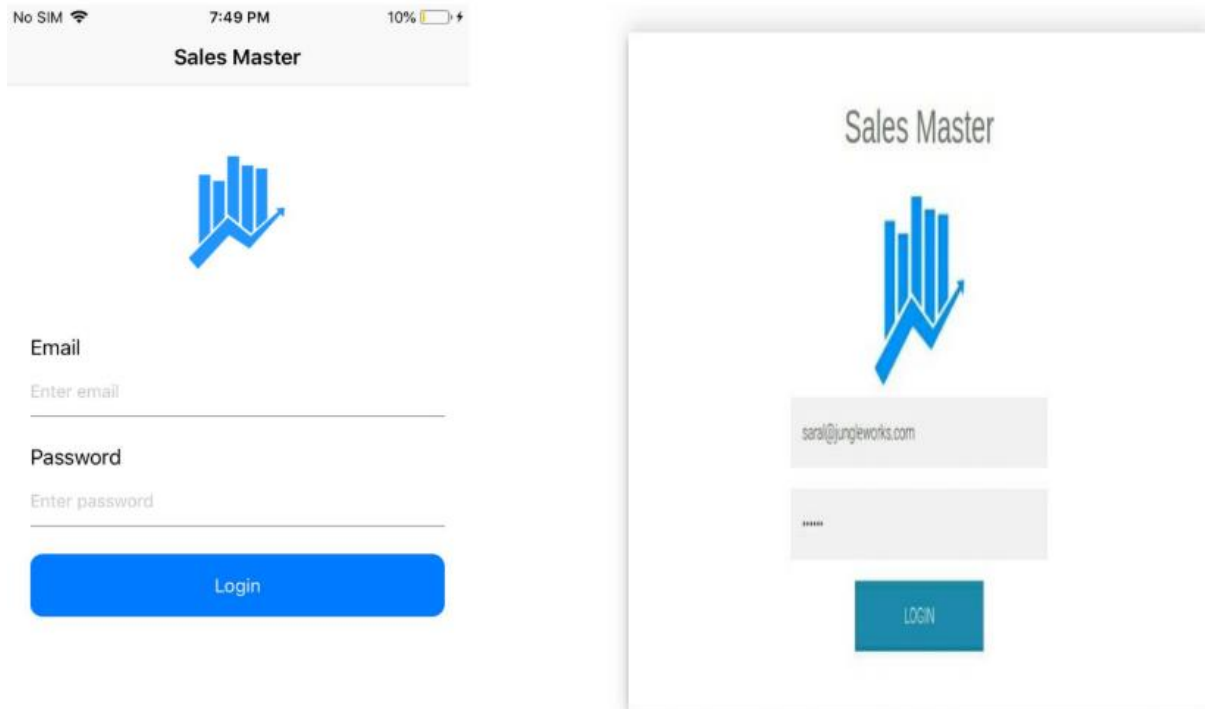


Fig 3.2 Login Profile

3.5.2 DASHBOARD:

The dashboard exclusively designed for the admin presents a list of all the enrolled teams along with the sales figure that correspond with them. All these details are fetched using get method to hit the database. In order to establish a sense of independence the database is accessed through API's and not directly, thereby using the API as the service layer. In order to provide extended management of the enrolled teams, the application includes a delete view that allows the admin to analyse and remove a team as per his own discretion.

Team's Revenue		Add
Rubal	\$ 71,380	
Sue	\$ 46,863	
im	\$ 28,942	Delete
Vikas	\$ 21,288	
Omkar	\$ 21,186	
Ashish	\$ 16,129	
Tushar	\$ 14,309	
Upasana	\$ 10,525	
CSM	\$ 2,800	
TOTAL SALE		\$ 233,422

Sales Order		Options
Sankit	\$670,789.00	Log Out
Aman	\$1,234.00	
Rohins Singh	\$100.00	
Dhruv	\$1,555,555.00	
Dulad	\$670,054.00	
Tookan	\$8,974,651.00	
Total Sale		\$11,113,793.00

Figure 3.3 Dashboard

3.5.3 ADD TEAM:

Another API is involved to insert validated inputs into the database i.e. name of the team along with the projected revenue.

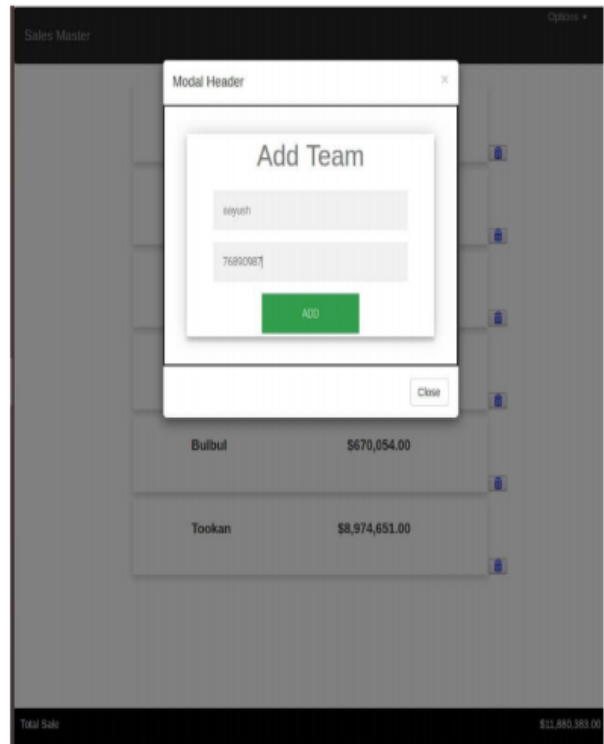
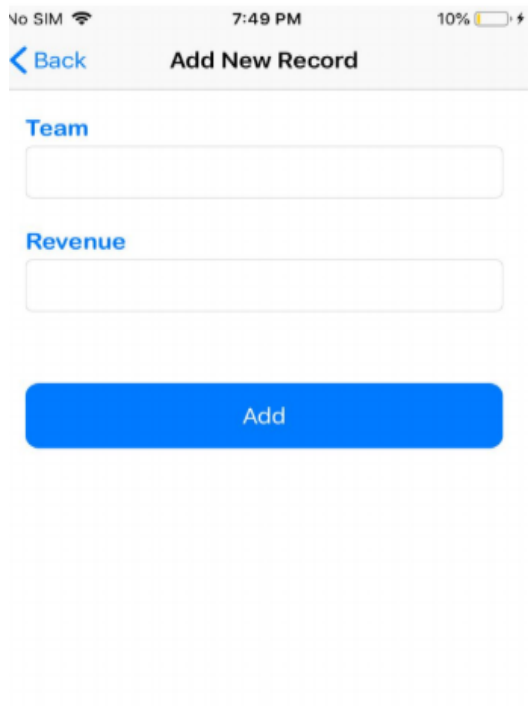


Fig 3.4 Add Team

3.5.4 UPDATE TEAM:

Once the record has been inserted, there are chances that admin might feel the necessities to update the credentials based on the login module supplied within the application. In order to facilitate this functionality, an update view has been included to update the details as per requirements.

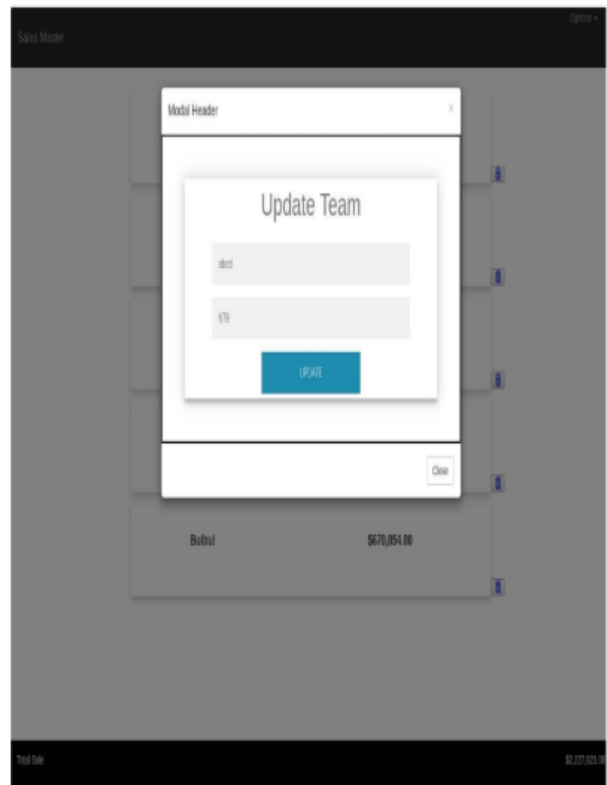
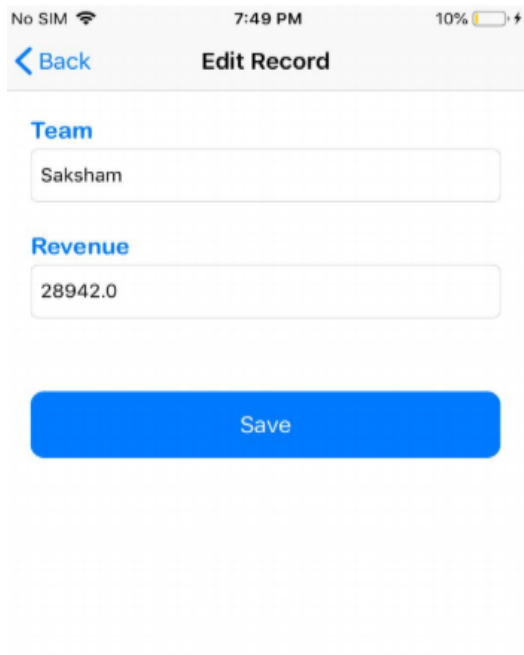


Fig 3.5 Update Team

3.6 COMMANDS:

In this project we have to run different commands in order to achieve successfully what we really want in our SHARED APPLICATION . Since we have to also run the web app too so we have to run some commands which we used in above project.

The list of commands are :

- In order to create a new shared project through command we can use the command under @nativescript/schematics as follows

ng new --collection=@nativescript/schematics project-name --shared

- *Web Build* builds a web app from the code-sharing project and opens it in default browser

ng serve -o

- *Mobile Build*

tns run ios --bundle

tns run android --bundle

3.7 CODE SPLITTING :

Re-usability of code is important principle of coding that allows similar modules to be dependent and in relation of correspondence. This also enables the programmer to reuse the code for multiple platforms. However this reuse might cause some errors due to incompatibility among the platforms, so in order to facilitate successful execution multiple copies of the same have to maintained as per the correspondence of the platform.

These are the files which are created for mobile app.

file.tns.ts : File for logics to be written

file.tns.html : File for document to be shown on screen

file.tns.css : File for designing

CHAPTER : 4

WORK PERFORMANCE

4.1 Description of Work Performed :

The major technology of application involved in this project is Angular. The choice of the use of this technology is its ability of reliable yet simple integration. This also involves dependency injection, template in order to solve development challenges. Angular bridges the gap between the platforms enabling the developer to develop a unified application available across all platforms namely web, mobile and desktop. The project work has empowered me to understand the basic outline of MVC, Angular in order to formulate a website successfully able to function as per the proposal presented to the developer by the client.

4.2 Major Challenges :

One of the main challenges observed by me as a developer was introduction to angular for MVC. The major challenge was my ability to establish an efficient front end that shall stand successful in appeal as well as simplicity to interacting user. The extension of the application through use of CSS to add further more appeal as well as adaptability to the web site was further more a challenge. Another challenge for as a newbie was the ability to make my code as much community friendly as possible to follow the company guidelines which shall improve induction of new team by making it simpler for them to understand the code and to understand the project in order to scale it up.

However, the project motivated me to push myself against my limit and improve my output inch-by-inch.

4.3 Key Learnings:

The key learning not only included the knowledge of the technology involved but also the integration of cross desktop development through GIT that facilitated developers to merge their contribution into a single unified project.

CHAPTER : 5

CONCLUSION

In general, I would portray my entry level position as a positive and informational experience. There were a few issues amid the principal week, as it was an entirely different field for me. Along these lines, it set aside some effort to make sense of first and foremost lastly begin with it. It was motivating to perceive how a portion of the engineers had the option to for the most part freely complete their interests and pla and play out their very own analyses. It was likewise fascinating to encounter a totally different IT condition, where numerous engineers were contributing their best and helping each other out to accomplish long and transient objectives set by the organization. Then again, getting an opportunity to chip away at a live venture was another energizing and testing knowledge. In the event that I need to total up my entry level position in one line, I would state this was a strong beginning of my profession.

REFERENCES

The material that we have referred that helped us with the development of our Angular POC project are as follows:

1. <https://angular.io/guide/quickstart>
2. <https://docs.angularjs.org/tutorial>
3. <https://material.angular.io/>
4. <https://github.com/johnpapa/angular-styleguide>
5. <https://docs.nativescript.org/>
6. <https://docs.nativescript.org/angular/start/introduction>