INTERNSHIP REPORT ON TOPIC NAME

AT

ALFRED TECHNOLOGIES (OPC) PVT. LTD.

FROM (1-10-19) TO (1-12-19) IN HYDERABAD

SUBMITTED BY - NAME

B.VOC (SOFTWARE DEVELOPEMENT) Vth SEMESTER

ROLL NO- 2702

Table of Contents

Acknowledgement

The internship opportunity I had with Alfred Technologies (OPC) Pvt. Ltd. was a great chance for learning and professional development.

Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it.

I am also grateful for having a chance to meet so many wonderful people and professionals who led me though this internship period. Bearing in mind previous I am using this opportunity to express my deepest gratitude and special thanks to **Mr. Hercules Singh Munda** of Alfred Technologies (OPC) Pvt. Ltd. for taking part in useful decision & giving necessary advices and guidance and arranged all facilities to make life easier.

I perceive as this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future,

Sincerely,

Abdul Basit

About Alfred Technologies (OPC) Pvt. ltd.

Alfred Technologies is a social technology company focused on improving society and enabling people for better outcomes across multilingual learning, cross-cultural experiences and diversity education.

Company leverage technology and human capacity and intellectual insights to deliver integrated solutions in order to influence social processes.

Company are reimagining the human experience of working with social issues.

About Universal Studio

Universal Studio is a online match making platform with operations in specifically in Universal. With a need to find match within the community, internet is the tool for connectivity.

The platform localized information system makes user flow seemingly smooth and easy. This also build up the trust toward the brand.

We leverage technology and human expertise to maintain the integrity of traditional values. The strong technical team ensure that no cases of fraud or misleading goes undetected through the system.

How it Works

We collect data and different parameters and run analysis using data science and provides user their best matches.

Vision

To Develop Toursim & art & culture in Universal in accordance with Digital India moment.

Technology Stack

Technology used are as follows:

- React-Native (Front-end) with Expo (Managed workflow),
- Redux (State Management Library),
- React-Navigation (Handles Navigation as per Material design in react-native),
- Django python Framework (Back-end),
- SQLite 3 (DBMS)

Brief About Each Technology

React-Native - React Native combines the best parts of native development with React, a best-in-class JavaScript library for building user interfaces.

React Native lets you create truly native apps and doesn't compromise on your users' experience.

It provides a core set of platform agnostic native components like *View*, *Text*, and *Image* that map directly to the platform's native UI building blocks.

React components wrap existing native code and interact with native APIs via React's declarative UI paradigm and JavaScript. This enables native app development for whole new teams of developers, and can let existing native teams work much faster.

Redux - A predictable state container for JavaScript apps, Enables you to manage centralized state, as single source of truth,

features - Centralized, Predictable, Large community, Debuggable and Flexible.

In redux we have,

React-redux - Allows you to dispatch action throughout the App by wrapping the component in HOC named Connect.

Redux-persist - Allows you to Store Redux state as persistent State, for example - while login in user store the token in Persistent storage, so that

when user comes back he will be logged in automatically and saves his effort and imporve UX/UI.

Redux-thunk - It provides a middleware to async action in redux.

React-Navigation - Open source project provides Handling Navigation For react-native.

features - Easy-to-use, Components built for iOS and Android, Fully Customizable, Extensible platform.

Django - Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

SQLite 3 - SQLite is a relational database management system contained in a C library. In contrast to many other database management systems,

SQLite is not a client–server database engine. Rather, it is embedded into the end program.

Architecture

Micro-Service Architecture

In Micro-Service architecture each segment/ Service divides in micro-service or we can say feature, every service is independent of other service in terms of performance and maintainability

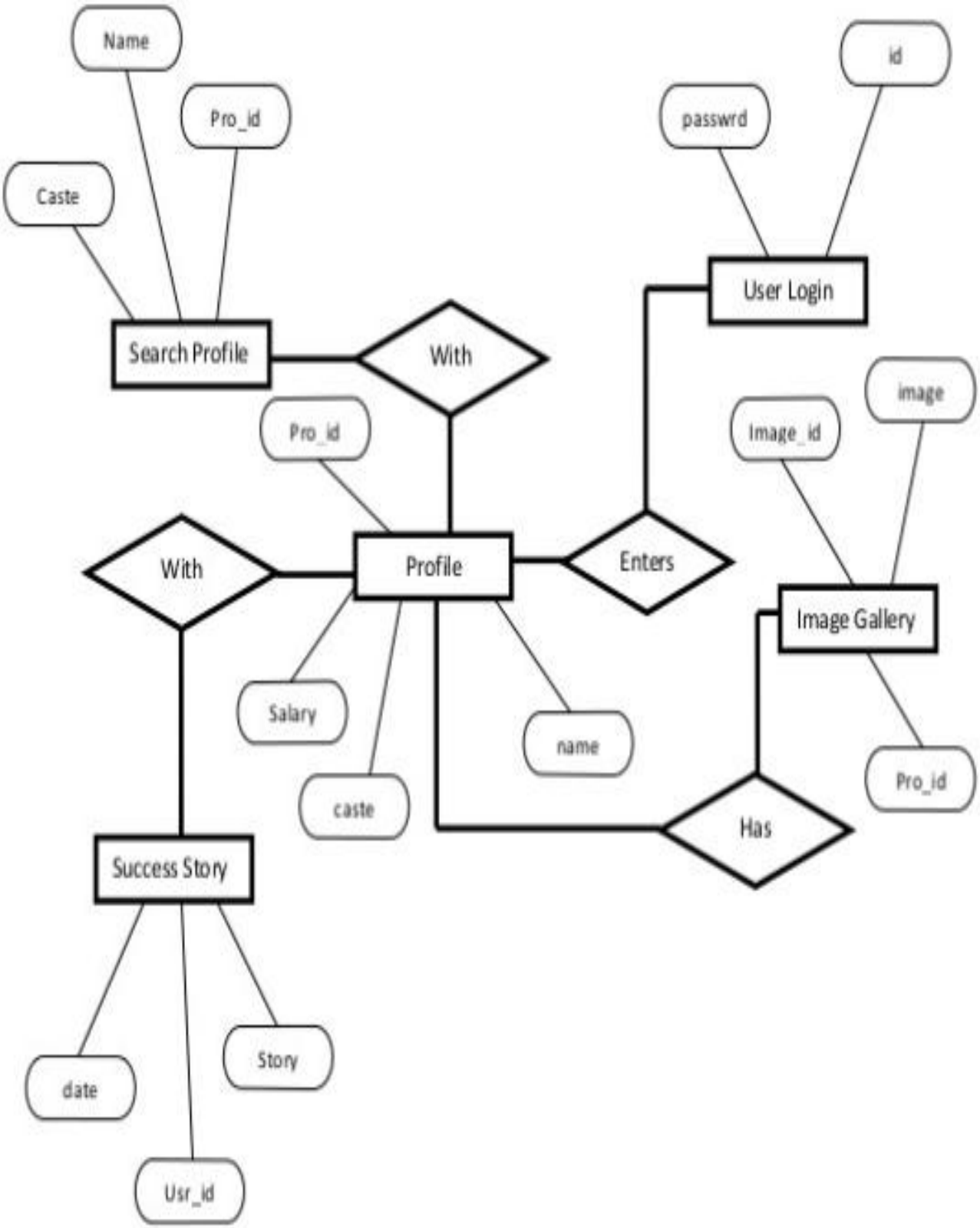
Each service or feature communicate with each other with REST API,

For example - Login, User Registration, Profile, each service is a micro-service,

Features of Micro-Service Architecture:

- Decentralized Governance
- Decentralized Data Management
- Infrastructure Automation
- Maintainable
- Flexible
- Easy Upgrades
- Easy Migration
- Everything is like Plug and Play.

ER Diagram


Name
Pro_id
Caste
Search Profile
With
Pro_id
With
Profile
Enters
Salary
caste
name
Success Story
date
Story
Usr_id
id
passwrd
User Login
image
Image_id
Image Gallery
Pro_id
Has

Weekly Overview of my activities

Week 1

In week 1 my objective is to make a basic Components to build the App (screenshots on next page)

- Fab button
- text field as Per UI
- Picker
- Profile Card
- Interactive Buttons

After the completion of basic component for App Screens,

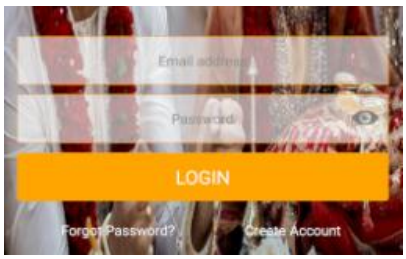
Screens development started (screenshots on next page)

- Home Screen
- Interest Screen
- Forgot Password Screen
- Search Modal Screen

Screenshots of components

Floating Action Button

Text Field as Per UI



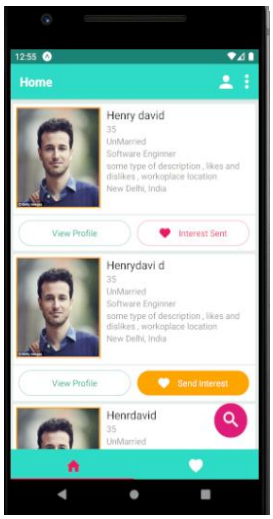
Profile Card

Interactive Buttons

Before Touch

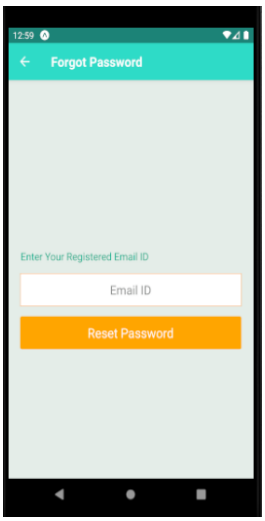
After Touch

Screens

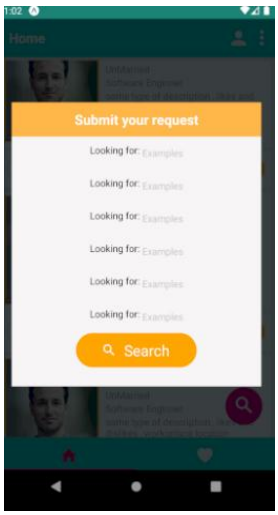


Home Screen

Interest Screen



Reset password screen



search Modal screen

Week 2

Continuing with screens

- Profile Screens

- Interest Sent Screen

- Interest Received Screen

combining Screens with components made before i.e., Integrate profile card and interactive buttons in Home Screen, Adding Fab with Home Screen and adding search Modal with fab button

Navigation throughout the App

It takes around 4 days to setup with a lot of reimplementation's

First used Drawer Navigation (Screenshot on Next Page) but after it is decided to make it more interactive using swipe navigation (Screenshot on Next Page)

reimplementation of navigation contains

- Stack Navigation

- Material Top Bar Navigation

- Switch Navigation

Issues faced during Navigation

Implementation of nested Material Tab navigation inside stack navigation giving hard hands

How I Resolved?

stack overflow and documentation overview, then restructured the final Navigation

Week 3

Developed Screens

- Login Screen

- Register Screen

Login screen is done in around one and half day,

In login Screen Password Text Input has to be made such user can view Password after Typing by clicking on side touch-eye (Screenshot on Next Page)

Register screen is a tough task because we had to take maximum information from user and also had to decrease the effort of user while input,

so initially there is 5 swipe able register screens, but as the App grows the complexity grows gradually, and worsen the performance of JS Thread around -2 fps

So this issue, I resolved using Modal component of react-native and dividing the Register user screen in to one screen of register then after the credentials verification navigate to the profile screen which has the rest of screens namely

-Basic Details,

-Personal Details,

-Professional details,

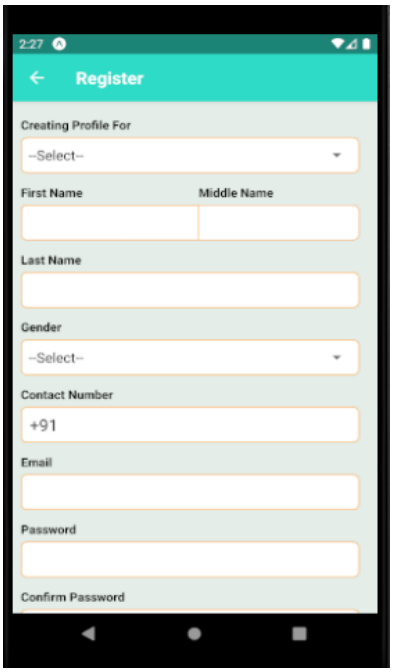
- Family Details,

-Preference Details and

-Invites suggestions

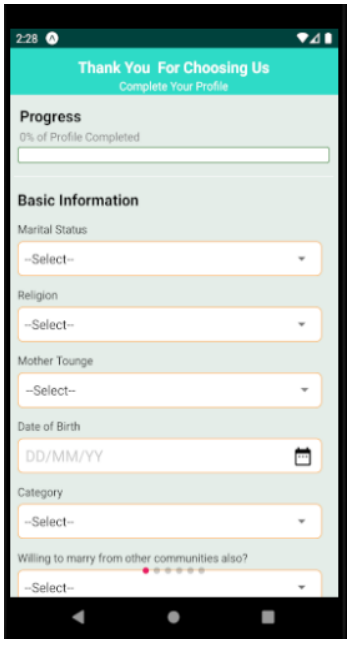
Which gradually improves the rendering time and Increase the performance of JS thread to normal of 60 fps

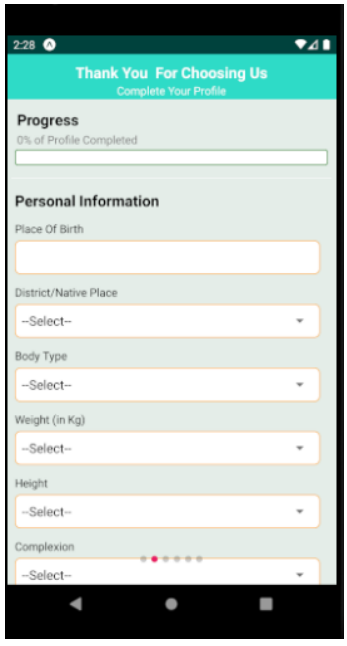
Screenshot of Screens

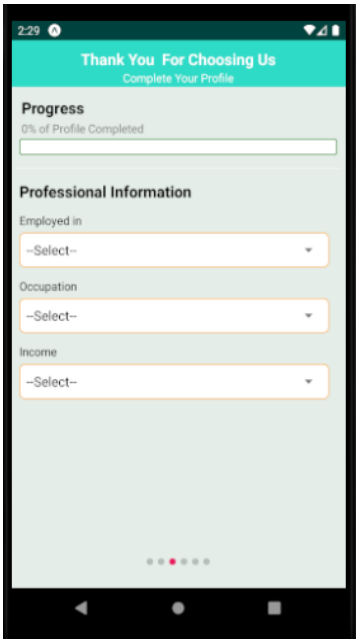


Login Screen

Register Screen





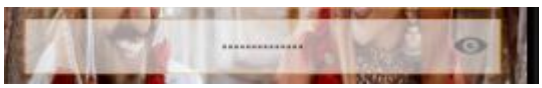


Basic Information Screen

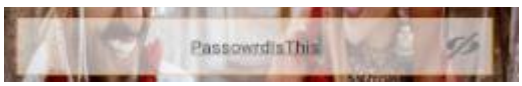
Personal Information Screen

Professional Information Screen

Show Password Input Component



Before



After

Week 4

These profile Section includes mainly 3-4 components,

Documented Components namely

-Picker List

-Text Input,

Manually developed Components

-Multi Picker List,

-Picker With section it takes around 2 days to complete with perfection

And now comes the

-Preference Screen,

-Invite Screen

In Preference Screen, I had to implement Photo picker using Camera Picker API and Camera Image Picker API

It has to be validated as maximum of 5 photos,

then I developed a Add photo component

then Implemented logic and validation for Image picker. (Screenshot in Next Screen)

In Invite Suggestion Screen, my task was to prompt the user to upload any number of contact (Minimum 2 contact is required) from contacts list of phone in order to know/help/suggest them about the app

Issue Faced -

API for this task is not available all over the Open Source Community

so, I had to develop the API using Contact API in order to Implement

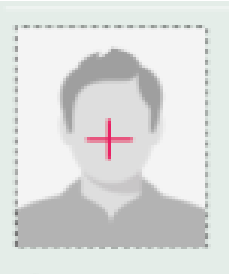
Contact Picker API

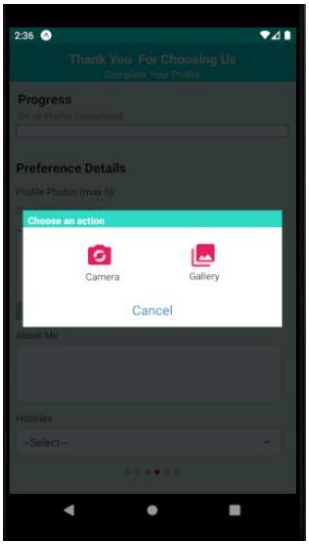
How I resolved -

First get All the contacts from the phone then ask user to select particular contact they want using FlatList and TouchableHighlight API (Screenshot on Next Page)

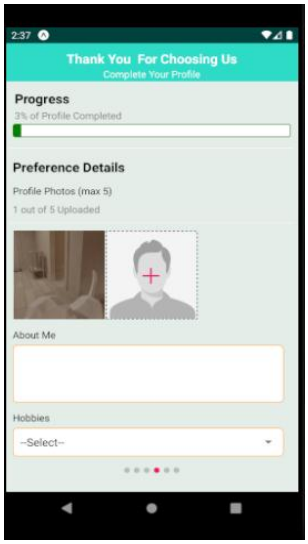
Screenshot of Components

Photo Picker Component

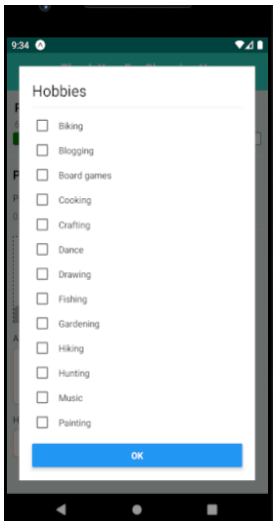




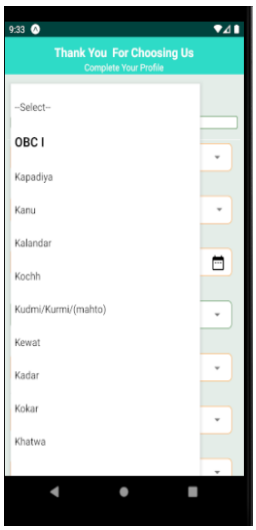
Preference Screen



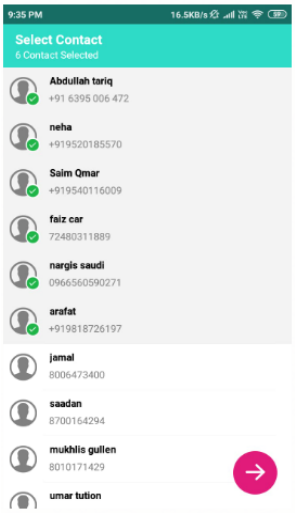
Multiple Picker List



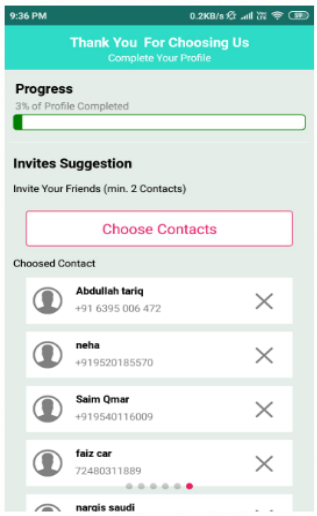
Picker with Section



Contact Picker

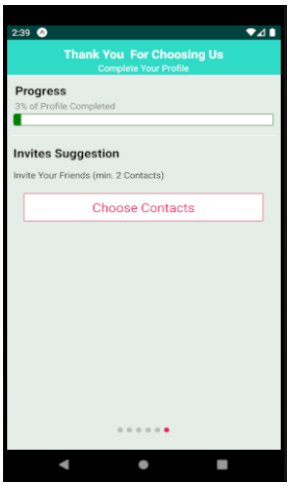


Contact Picker Result



Family Details Screen

Invite Suggestion Screen



Week 5

In week 5, I have Completed Validation for fields in Create Profile Section and Register Section and improve UX/UI

After that the process of integrating Backend comes, I have setup Redux, a state Management Library widely used with React/React-native and many other Cross Platform Frameworks, Redux Provides a Single Source of Truth for state, which can be easily accessible from any Component, no matter how deeply nested it is,

And to Provide Persistent Storage for the user, I setup the redux-persist.

Here are the some action Creators in my redux store

- logIn() async action which uses Middleware to dispatch an asynchronous action and responsible to set Token and error handling with persist storage, and also handle login API

- signUp() as async action which uses Middleware to dispatch an asynchronous action and responsible to set Token on Register and error handling with persist storage, and also handle SignUp API

Features To be Implemented

- ❖ Admin Panel
- ❖ User verification
- ❖ Membership Integration
- ❖ Payment Integration
- ❖ Real-time chat with Match
- ❖ Push Notifications
- ❖ Smart Search
- ❖ Add Horoscope
- ❖ Check Horoscope for the Particular profile
- ❖ Data Science Integration
- ❖ Promotions
- ❖ Social share
- ❖ Save Favourite

Technology Stack to be used

- Django framework
- SMS, Voice and Phone Verification - GupSup.io, Twilio, etc.
- Push Notifications – Twilio, Push.io, etc.
- Payments – PayTm, GooglePay etc.
- Real-time Analytics – Hadoop, Spark, BigData, Apache Flink, etc.
- Cloud Environment – AWS, Google

References

- React-Native Docs - https://facebook.github.io/react-native/docs/getting-started
- Redux Docs -https://redux.js.org/introduction/getting-started
- Django Docs -https://docs.djangoproject.com/en/2.2/
- Alfred technologies website - http://alfredtechnologies.in/