DOCUMENTATION and ACTION PLAN
HPDF T41

1. Zomato Web App

1.1. Google Statistics reflect the decreasing number of new mobile applications being install per month. This is a clear indication that the world is moving towards implementing their existing applications as Progressive Web Apps.

1.2. Here are some of statistics:

1.2.1. Pinterest rebuilt their mobile site as a PWA and core engagements increased by 60%. They also saw a 44% increase in user-generated ad revenue and time spent on the site has increased by 40%.
1.2.2. BookMyShow's PWA takes less than 3 seconds to load and increased conversion rates over 80%. The PWA is 54x smaller than Android and 180x smaller than iOS app.
1.2.3. Flipkart's PWA is driving 50% of its new customer acquisition. Nearly 60% of visitor to the PWA had previously uninstalled the native app primarily to save space.

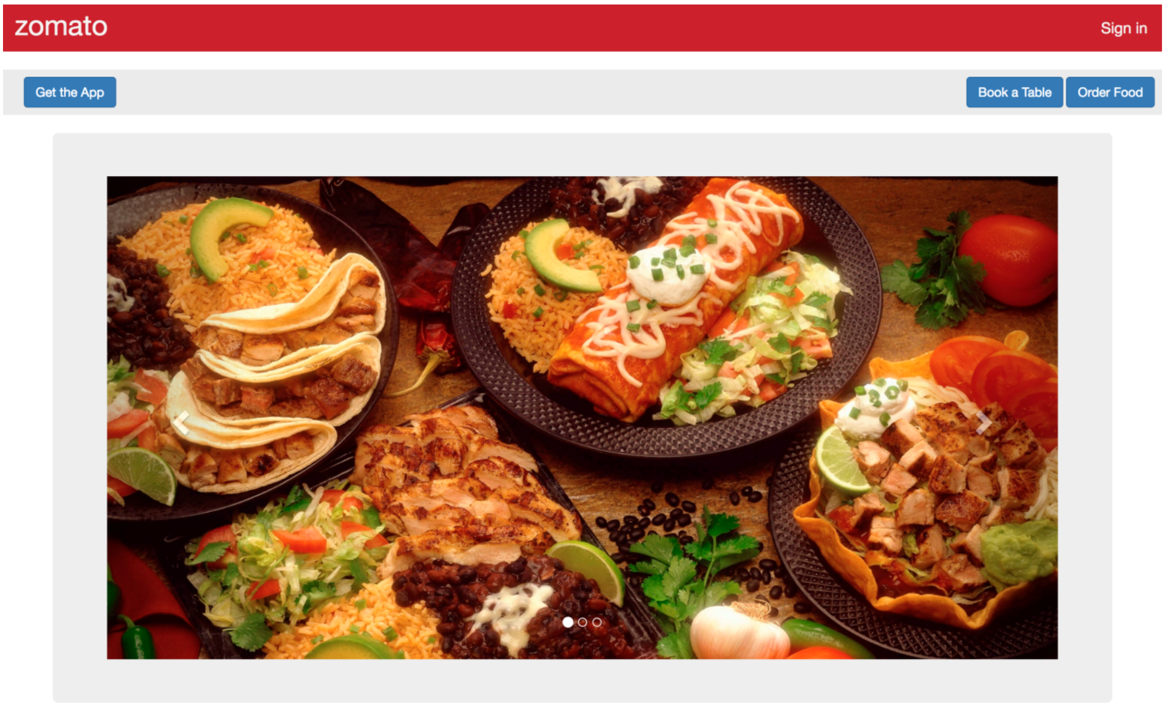
1.3. Keeping up with the trend, it is just advisable that our application takes the route of PWA as well. This would not only increase the boot time for the app, but can be also used in the offline mode by caching data. PWAs would have support for push notifications and take up zero to minimal space on phone.

1.4. Our team (T41) has been divided into two groups of four members: PF and NE. While the members of the PF would be working on the Web-app, the members of the NE group would be designing and developing the Android application.

1.5. We have to keep in mind that installable-mobile applications are not completely redundant yet. This requires us to develop for mobile OSes as well.

1.6. Updates 11th January 2018

1.6.1. Currently we are working on the front-end for the applications, i.e., wireframes. The language/framework for the main structure of the website is Python/Flask. This makes the website very light and easy to load at client's end.

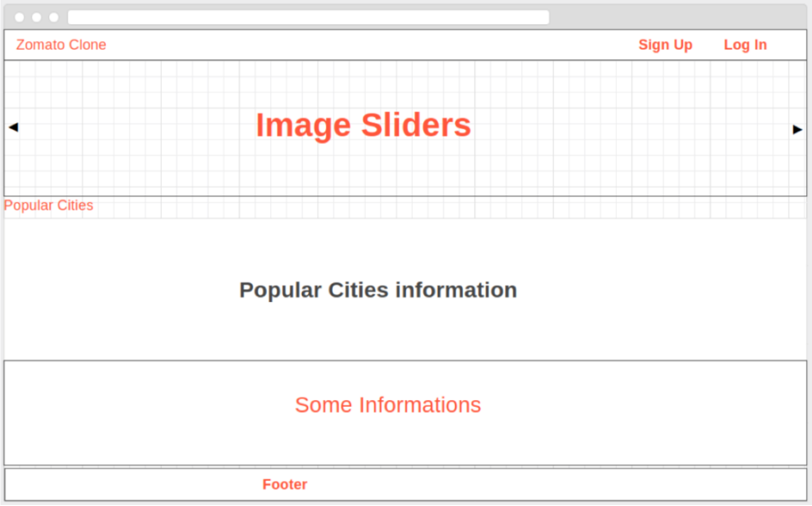


1.6.2. This is being followed by developing the Sign In, Order Food and Book a Table pages which are to linked to the buttons present on the front page. Once an MVP of the Android application is created and an apk file generated, Get the App button would be enabled to link to the repository where the application is stored.

1.6.3. A light structure of the website would be deployed to Hasura's cluster and users can access if they have the URL to it.

1.7. Plan of Action:

1.7.1. Landing Page:

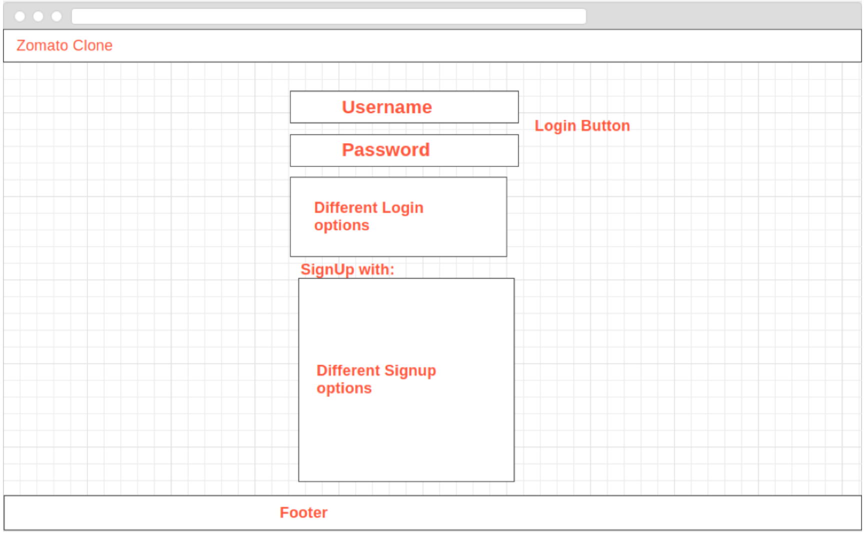


1.7.1.1. The landing page consists of the signup and the login options in the header bar of the page. Below the Carousel would be the display of information regarding restaurants in the user's current city.

1.7.1.2. The city would be auto-detected or can be entered by the user from a drop-down list of available cities.

1.7.1.3. The footer of the front page will be showing some useful links such as major cities, contact us, etc.

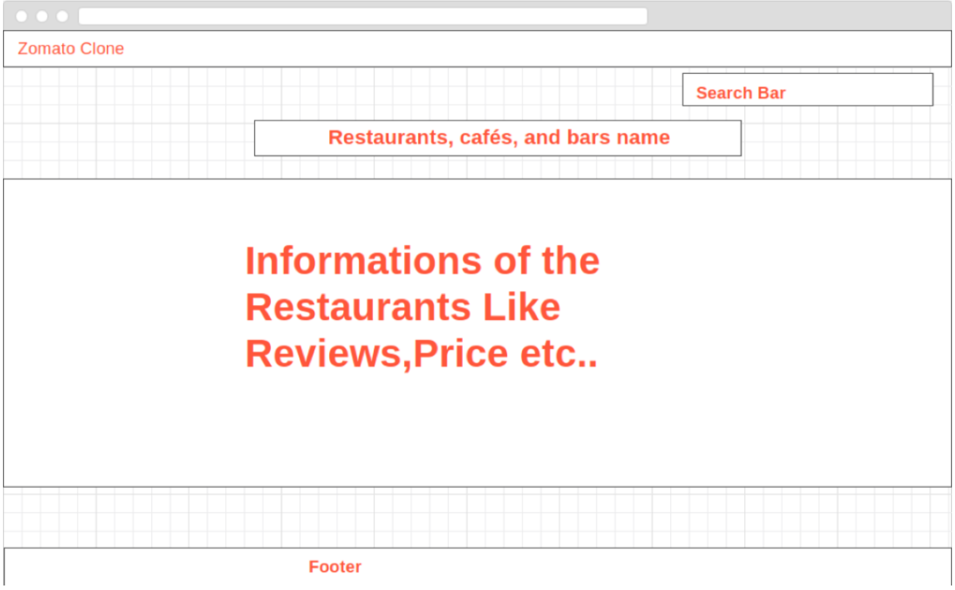
1.7.2. Login/Signup Page:



1.7.2.1. The signup and login options through different social media websites can be implemented by Hasura Authentication System. We would be providing the following options:

1.7.2.1.1. Signup/Login with Facebook
1.7.2.1.2. Signup/Login with Google
1.7.2.1.3. Signup/Login with Twitter

1.7.3. Main Page:


Zomato Clone
Search Bar
Restaurants, cafés, and bars name
Informations of the Restaurants Like Reviews,Price etc..
Footer

1.7.3.1. The main page would be an extended scroll webpage for ease of use. A search bar would exist at the top to the page. Users may click on different restaurants to check out their menu, location, reviews by other people (to be implemented).

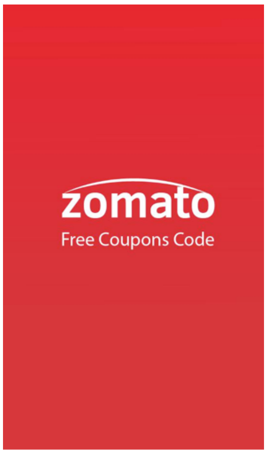
1.8. Technologies Used:

1.8.1. Front-end : ReactJS
1.8.2. Back-end : Python/Flask, NodeJS

2. Android App Plan of Action:

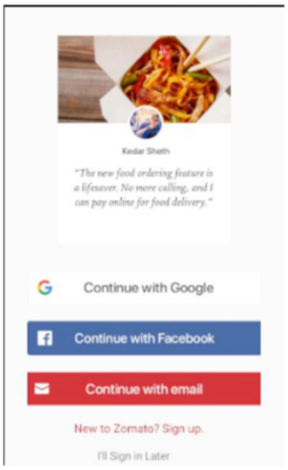
2.1. Boot Page

2.2. Users would be welcomed by a splash screen which would contain the app logo (made using LogoLauncher) and would have a desired timeout.



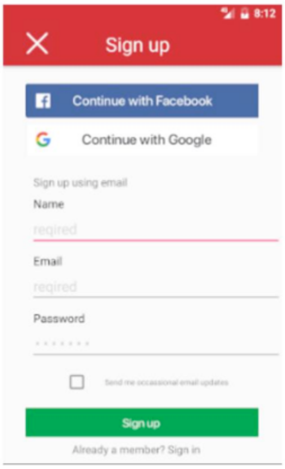
2.3. Sign-up/Sign-in page:

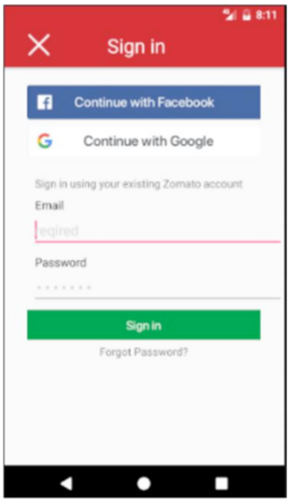
We will be providing the options for signing up using Facebook and Google. This can be later used to sign in.



Kedar Sheth

"The new food ordering feature is a lifesaver. No more calling, and I can pay online for food delivery."





2.4. Main Page:

The following options would be provided:
Dining Out
Delivery
Trending

2.5. These pages would be made using Tab navigation

2.6. Profile view:

Profile view provides:
Notifications
Bookmarks
Nearby
Home
Online orders

2.7. Profile view is done by Drawer-navigation.

2.8 Technologies Used:

Front-End: Expo xde, React Native
Back-End: Python/Flask, NodeJS

Database and APIs

We will be using the hasura cluster interface to create the database and will generate the API code as needed in the respective languages.

Plans on collaboration

Using Hasura CLI creating a common cluster and adding whole team members in that cluster as collaborators.
Similarly create a GitHub repository accessible by team members.
Adding everybody's project to GitHub and then to cluster as needed.

Time-Line of the Project:

- 14th January, 2018 : First Draft of Action Plan
- 4th February, 2018 : First Draft of Project on GitHub
- 11th February, 2018 : Submission on Hub
- Last week of February, 2018 : Submission of extended idea

Sources:

- Hasura's Documentation
- Flask Documentation
- Node.js Documentation
- React Native Documentation
- React.js Documentation
- Django Documentation
- Wikipedia
- W3Schools

-X-