**HPDF 2017-2018**

**Team -10**

**Final Docs for Submission -1**

**Node.Js – Express**

|  |  |  |
| --- | --- | --- |
| S.no | Backend | Hasura Features To Be Use |
| 1 | Providing backend for the Authentication page of the front end , storing passwords locally on the server. | Hasura Auth , Hasura Data instance, Postgres instance |
| 2 | Providing Backend frame work for storing user data in the app server | Hasura Data , Postgress Instance, Hasura File |
| 3 | Storing User Status (Logged in or logged out etc) | Hasura Data, Postgress Instance |
| 4 | Sending custom notifications , history of notifications | Firebase cloud messaging services, Hasura Data, Hasura File , Hasura Postgres instance |
| 5 | Notification Status Info | Hasura Data , Hasura Postgres instance |
| 6 | Error handling | - |
|  |  |  |
|  |  |  |

**Features of the Backend for the App are as follows.**

1.     Firebase cloud messaging service will be used for sending the notifications.

2.     Authentication Service using Hasura Auth

3.  Hasura Postgres instance for storing User basic information as well as the group info.

4.  Hasura Data API will be used for inserting and fetching data as requested by the front end app

5.  Hasura File API for sending files like images along with the notification.

**Hausas Features to be used:-**

1.     Data Storage: - Postgres instance in the cluster will be used for storage of the messages as history or records etc.

2.     Inserting Or Fetching the data from the DB :- Hasura Data API

3.     Hasura Auth will be used for providing the basic authentication services

4.     Hasura File Api will be used for storing static files like images etc which might be sent in the notification.

****

**Features for the Front End**

**React Js**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Sl no.*** | ***Frontend case*** | ***Expected Hasura Backend feature required*** |  |
|  | 1 | Login screen with user id and password box & a register button to add new users | Hasura auth & Hasura data API |  |
|  | 2 | Register screen with name, username, email and ph no. | Hasura data API |  |
|  | 3 | After login--User profile to add/edit user image, ph no. & email | Hasura file API |  |
|  | 4 | Home screen with regular and admin functions with user support integration(help button) | Hasura auth & Hasura data API |  |
|  | 5 | Topmost USERS box to show scrollable user list (showing online & offline users separately with a green dot ) | Hasura data API |  |
|  | 6 | Predefined GROUPS with a group msg button to display a list of groups , the user is a part of | Hasura data API |  |
|  | 7 | Select multiple users to create a new group | Hasura data API |  |
|  | 8 | Pop-up notification to display MOST RECENT received msg in 1 line or 100 characters as a summary header | Hasura data API |  |
|  | 9 | ALL Notifications button with a badge number for unread notifications | Hasura data API |  |
|  | 10 | On click- Display all sent,received & draft notification (different tabs) on clicking notif button | Hasura data API |  |
|  | 11 | Send to All button to send notifications for system wide changes or policy updates | Hasura auth & Hasura data API |  |
|  | 12 | Pop up dialog box to push a notification with send button | Hasura data API |  |
|  | 13 | Pop up dialog box to notify if the sending the notification was successful | Hasura data API |  |
|  | 14 | If not display a RETRY/resend button or edit the message before trying again | Hasura data API |  |
|  | 15 | Logout button on upper right corner | Hasura auth |  |

****

**Team Members (T10)**

* **Node Js –Express**
  + **Ankit**
  + **Mr Vittal**
* **React Js**
  + **Manish**
  + **Anandita**
* **React Native**
  + **Sreelekshmi**
  + **Vicky**
* **Python Flask**
  + **Kiran**

****