### 

### **Problem Statement**

Build an email client app like Outlook.

Following APIs are provided:

1. Get all emails
2. Get email body for a particular email

### **Sample UI**

[Email List View (http://bit.ly/2VtQGcb)](http://bit.ly/2VtQGcb)

[Email body View (http://bit.ly/2I5DemI)](http://bit.ly/2I5DemI)

[Color codes (http://bit.ly/2wa2pCa)](http://bit.ly/2wa2pCa)

### **Product Features**

1. The app features an email list page. This page shows the list of emails sent to a user.
2. Clicking on any email item in the list should split the screen into a master-slave (left-right) screen type where the master (left) shows the email list (with the selected email item) while the slave (right) shows the body of the email. The body of the email is not known ahead of time and should be loaded only when the email item is clicked.
3. The app should allow any particular email item to be marked as “favorite”, it should be done via clicking on an email item and then clicking the “Mark as Favorite” button in the email body section.
4. The app should show read and unread mails in different CSS styles to distinguish between the same.
5. Allow filtering emails by “favorites”, “read” and “unread”.

**Must Haves:**

* Render all emails page using the API
  + Each email should have from, subject, short description, date and time.
  + The avatar (circular logo) in each email item should be populated with the first character of first name (sent in API response).
* Upon clicking a particular email, render the body section for it using the API. Email body has 3 sections:
  + Email subject
  + Email body
  + Email date and time
* Allow email to be marked favorite in the body section of the email
* Filter emails marked as favorite, read and unread
* UI should be as close to the mocks provided
* The date should be rendered in format **dd/MM/yyyy hh:mm a**

**Good to Have:**

* Email list could be long and hence is paginated. There are 2 pages i.e. page 1 and page 2 which can be accessed via the APIs provided below.
* Persist favorited and read emails across sessions using persistent storage technologies

## **Points to consider:**

* Focus on the modularity of code and design of the solution. Keep performance of the application in mind.
* **The solution should be coded in React**. **You *can* use tooling such as webpack, grunt, gulp, babel etc**
* The final solution should work without errors.
* Do not completely ignore the layout / visual design. A minimalist visual design / layout must be followed, it is also important for us to evaluate your CSS knowledge.
* Do not create a div soup instead use semantic HTML tags

## **You will be evaluated based on:**

1. Correctness and completeness of the solution.
2. Code design and quality.
3. Visual aesthetics (the UI should be as close as possible to the given design).
4. Technology choices (e.g. ES 6/7 over ES 5 - avoid mixing of ES 5/6/7).
5. Your understanding of the problem statement.

### **API Sources**

1. Emails List APIs:

<https://flipkart-email-mock.now.sh/> (Not Paginated)  
[https://flipkart-email-mock.now.sh/?page=](https://flipkart-email-mock.now.sh/?page=1)<pageNumber> (Paginated)  
e.g. <https://flipkart-email-mock.now.sh/?page=1> and [https://flipkart-email-mock.now.sh/?](https://flipkart-email-mock.now.sh/?id=3)page=2

1. Email body API:   
   [https://flipkart-email-mock.now.sh/?id=](https://flipkart-email-mock.now.sh/?id=3)<email-item-id>

e.g. <https://flipkart-email-mock.now.sh/?id=3>