



WEB TECHNOLOGIES

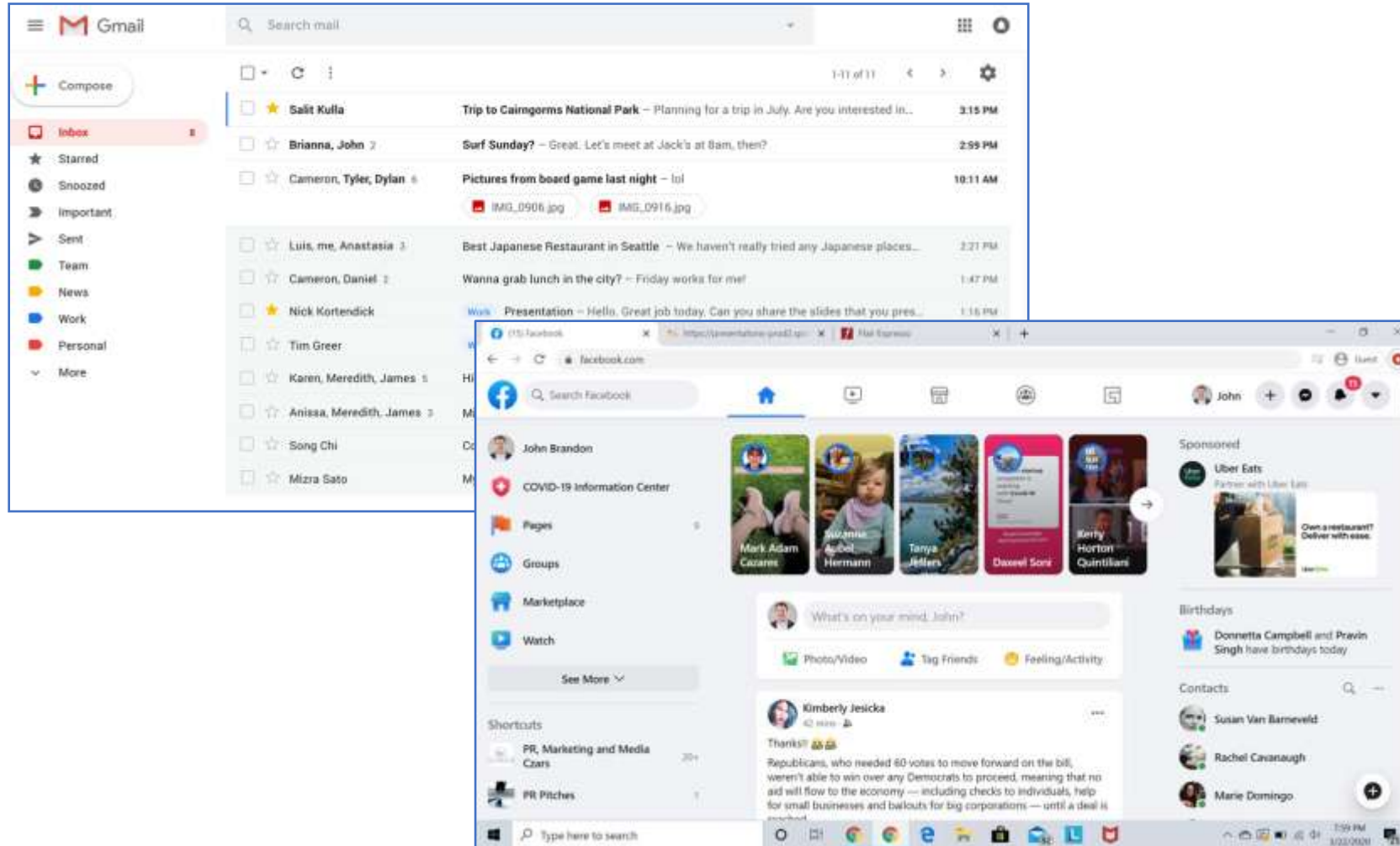
Single Page Applications & Asynchronous Communication

Vinay Joshi

Department of
Computer Science and Engineering

Single Page Applications

Introduction



- Instead of the default method of the browser loading entire new pages, a single-page application (SPA) interacts with the web browser by dynamically rewriting the current web page with new data from the web server
- Resources are dynamically loaded and added to the page as necessary, usually in response to user actions
- The page does not reload at any point in the process, nor does it transfer control to another page
- Can be built using
 - AJAX
 - Frameworks like ReactJS, AngularJS

- Traditional web applications, upon request from user like clicking a link or submitting a form, a new page is loaded with requested resources
- Asynchronous applications, upon user action, updates a part of the page without reloading the entire page
- Approaches include
 - Setting src property of iFrame or img element
 - A more elegant and complete approach is use of XHR or XMLHttpRequest object
- First create an XHR object using
var xhr = new XMLHttpRequest();

Asynchronous Communication

XHR object properties and methods

Properties / Methods	Description
<code>open(<i>method</i>, <i>url</i> [, <i>asynchronous</i>])</code>	Initializes the request in preparation for sending to the server. The <i>method</i> parameter is the HTTP method to use, for example “GET” or “POST”. The <i>url</i> is the relative or absolute URL the request will be sent to. The optional <i>asynchronous</i> parameter indicates whether <i>send()</i> returns immediately or after the request is complete (default is true, meaning it does not wait for response to come back)
<code>onreadystatechange</code>	Function to call whenever the readyState changes
<code>send([<i>body</i>])</code>	Initiates the request to the server. The <i>body</i> parameter should contain the body of the request, i.e., a string containing <i>fieldname=value&fieldname2=value2... for POSTs</i> or a null value for GET request

Asynchronous Communication

XHR object properties and methods

Properties / Methods	Description
readyState	Integer indicating the state of the request, either: 0 (uninitialized) 1 (loading) 2 (response headers received) 3 (some response body received) 4 (request complete)
status	HTTP status code returned by the server (e.g., 200, 404, etc.)
responseText	Full response from the server as a string (responseType property is set to “text” – default)
responseXML	A Document object representing the server’s response parsed as an XML document (responseType property is set to “document”)
response	Any other type of response received ((responseType property is set to “blob” or “json”)

Asynchronous Communication

XMLHttpRequest – Code Example



```
let xmlhttp = new XMLHttpRequest();
xmlhttp.open("GET", filepath, true);
xmlhttp.onreadystatechange=handler;
[xmlhttp.responseType="json"|"document"|"blob"] // default text
xmlhttp.send(null);

function handler() {
    if(this.readyState == 4 && this.status == 200) {
        // use this.response (json/blob) or this.responseText (text) or
        // this.responseXML (document) to update a part of the page
    }
}
```



THANK YOU

Vinay Joshi

Department of Computer Science and Engineering

vinayj@pes.edu

+91 80 2672 6622