

**Course Title:** Cloud computing Methodlogies

**Course Code:** CSI 3001

**Class Number:** VL2024250101880

**Semester:** Fall Semester

**Faculty:** Prof. Muthunagai S U

**Assessment:** Project

**Team members:**

* Shanmukha Ganesh S [21MID0063]
* Eshwar Reddy Y [21MID0085]
* Venkata Subrahmanya Deepak N [21MID0234]
* Eshwar Sharma Topudurti [21MID0237]

**API Topic:** Facebook API

**Title:** Echoes of Me! – Online Personal Gallery

**Report**

**Project Overview:**

"Echoes of Me!" is an innovative online personal gallery designed to empower users to curate and showcase their most cherished moments and creative expressions. In today’s interconnected world, creating an engaging and seamless user experience is paramount. To achieve this, the platform integrates the **Facebook API**, unlocking a suite of features and services that simplify usability, enhance interactivity, and foster social connectivity.

At the heart of this integration lies the **Facebook Graph API**, a robust and versatile tool that enables the seamless exchange of data between the platform and users’ Facebook accounts. Through this, "Echoes of Me!" ensures users can effortlessly share their gallery content, import memories, and connect with their social circles without navigating complex processes.

It is an web application that runs both offline as well as online using Facebook API. It is basically a personal hub of pictures of yourself posted on Facebook or Facebook owned products, a place where you can find all the posts you liked so that you can download with all the privacy and security that is needed.

**Facebook API:**

The Facebook API, also known as the Meta API, is a set of tools and services provided by Meta (formerly Facebook) that allows developers to integrate their applications with Facebook's platform. These APIs offer functionality for interacting with Facebook's social graph, accessing and managing user data, creating posts, managing ads, and more. The Facebook API, now part of Meta's suite of developer tools, enables integration of various Facebook services into your website, enhancing user engagement and functionality.

Its **Key Features** are:

1. Facebook Login:

* **Purpose**: Enable users to log in to your website using their Facebook credentials.
* **Use Cases**:
* Simplified registration/login process.
* Retrieve user profile data (with permission).

1. Social Plugins:
   * **Purpose**: Add social media features to your website without coding them from scratch.
2. Graph API:
   * **Purpose**: Access and manage Facebook data programmatically.
   * **Features**:

* Fetch user profile details.Post content to a Facebook Page or user feed.
* Retrieve posts, likes, and comments.

1. Sharing on Facebook:

* **Purpose**: Let users share website content on Facebook.
* **Options**:
* Use the Facebook Share Dialog.
* Predefine content, links, and images for sharing

1. Marketing API:
   * **Purpose**: Manage Facebook Ads and marketing campaigns from your website or backend.
   * **Capabilities**:

* Create, update, and monitor ads.
* Retrieve insights on ad performance.

1. Messenger API:

* **Purpose**: Integrate Facebook Messenger on your website.
* **Use Cases**:
* Enable customer support via Messenger chat.
* Automate responses using bots.

1. Insights and Analytics:

* **Purpose**: Monitor and analyze traffic and user interaction from Facebook to your website.
* **Use Cases**:
* Track the effectiveness of Facebook ads and posts.
* Measure engagement using events (e.g., page views, clicks).

1. Instagram Integration:
   * **Purpose**: Manage Instagram Business accounts via the Facebook API.
   * **Use Cases**:

* Post to Instagram from your website.
* Retrieve media and insights.

**Facebook API Services Implementation in our Project:**

* **Facebook Login [Public\_profile]:** Simplifies account creation and **login** by allowing users to **authenticate with their Facebook** credentials. Eliminates the need for manual form-filling, making onboarding swift and user-friendly.

window.fbAsyncInit = function () {

window.FB.init({

appId: '1307986103553292', // Replace with your actual App ID

cookie: true,

xfbml: true,

version: 'v16.0', // Latest valid API version

});

setSdkLoaded(true);

console.log('Facebook SDK Initialized');

const token = localStorage.getItem('fb\_access\_token');

if (token) {

validateAccessToken(token);

}

};

(function (d, s, id) {

let js, fjs = d.getElementsByTagName(s)[0];

if (d.getElementById(id)) return;

js = d.createElement(s);

js.id = id;

js.src = '<https://connect.facebook.net/en_US/sdk.js>';

fjs.parentNode.insertBefore(js, fjs);

})(document, 'script', 'facebook-jssdk');

}, [navigate]);

const validateAccessToken = (token) => {

window.FB.api('/me', { access\_token: token }, (response) => {

if (response && !response.error) {

setAccessToken(token);

fetchUserInfo();

onLogin(); // Notify parent component of login

navigate('/upload');

} else {

console.error('Invalid or expired access token. Logging out...');

localStorage.removeItem('fb\_access\_token');

setAccessToken(null);

}

});

* **Content Sharing [Users\_posts]:** Empowers users to share their curated galleries or individual pieces directly on their Facebook timelines. Drives engagement by allowing their friends and followers to view, like, and comment on shared content.

window.FB.api(

'/me/posts',

'GET',

{ access\_token: accessToken, fields: 'message,created\_time,full\_picture' },

(response) => {

if (response && !response.error) {

setPosts(response.data || []);

} else {

setErrorMessage(

response.error ? response.error.message : 'Error fetching posts.'

);

}

}

);

* **Personalization Through Graph API [User\_photos]:** Users can import selected photos, albums, or videos from their Facebook profiles to enrich their galleries. Makes **content management intuitive** by leveraging existing social media assets.

<div className="mainpage">

<h1>Your Facebook Posts</h1>

{errorMessage && <p style={{ color: 'red' }}>{errorMessage}</p>}

{!errorMessage && posts.length === 0 && <p>No posts available to display.</p>}

<div style={{ display: 'flex', flexWrap: 'wrap', gap: '16px' }}>

{posts.map((post, index) => (

<div

key={index}

style={{

border: '1px solid #ccc',

padding: '16px',

borderRadius: '8px',

width: '300px',

boxShadow:'0px 1px 1px black'

}}

>

<p>

<strong>Message:</strong> {post.message || 'No message'}

</p>

<p>

<strong>Created:</strong> <span className="layout">{new Date(post.created\_time).toLocaleString()}</span>

</p>

{post.full\_picture && (

<img

src={post.full\_picture}

alt="Post"

style={{ width: '100%', borderRadius: '8px' }}

/>

)}

</div>

))}

</div>

</div>

* **Analytics and Insights:** By integrating the **Facebook Pixel**, the platform tracks user engagement and provides insights to **improve usability**. Helps users understand how their galleries are being viewed and interacted with by their audience.
* **Location Feature [Users\_location]:** The Graph API can retrieve location data that users explicitly include in their Facebook posts, such as **tagged places**. Example: If a user posts a photo and tags "Central Park," the API can fetch that location data. While the Facebook API itself **doesn't offer direct geolocation services**, its ability to interact with location-related data can be combined with other tools to create a powerful, location-aware application.

useEffect(() => {

const accessToken = localStorage.getItem('fb\_access\_token');

if (!accessToken) {

setErrorMessage('Access token not found. Please log in again.');

return;

}

// Fetch user's posts

window.FB.api(

'/me/posts',

'GET',

{ access\_token: accessToken, fields: 'message,created\_time,full\_picture' },

(response) => {

if (response && !response.error) {

setPosts(response.data || []);

} else {

setErrorMessage(

response.error ? response.error.message : 'Error fetching posts.'

);

}

}

);

}, []);

useEffect(() => {

// Filter posts based on the category

const filterPosts = () => {

switch (category) {

case 'Family':

return posts.filter(post =>

post.message?.toLowerCase().includes('family')

);

case 'Friends':

return posts.filter(post =>

post.message?.toLowerCase().includes('friend')

);

case 'Places Visited':

return posts.filter(post =>

post.message?.toLowerCase().includes('visited') ||

post.message?.toLowerCase().includes('travel') ||

post.message?.toLowerCase().includes('trip') ||

post.message?.toLowerCase().includes('visit')

);

case 'About Me':

return posts.filter(post =>

post.message?.toLowerCase().includes('about me') ||

post.message?.toLowerCase().includes('bio') ||

post.message?.toLowerCase().includes('i am') ||

post.message?.toLowerCase().includes('me')

);

default:

return posts;

}

};

setFilteredPosts(filterPosts());

}, [category, posts]);

**Conclusion:**

The integration of the Facebook API not only streamlines website functionality but also enhances the overall user experience. Users no longer need to upload, manage, or promote their content across multiple platforms manually. Instead, they benefit from automated, efficient processes that keep their personal gallery dynamic and socially connected. By leveraging familiar tools and services, "Echoes of Me!" bridges the gap between personal expression and social interaction, ensuring a platform that is as intuitive as it is engaging.

In essence, the Facebook API transforms "Echoes of Me!" into more than just a gallery—it becomes a social hub for creativity and connection, empowering users to express themselves effortlessly while staying connected to their social networks.

**Code:**

**App.jsx:**

import React, { useState, useEffect } from 'react';

import { BrowserRouter as Router, Route, Routes } from 'react-router-dom';

import Navbar from './Navbar';

import Login from './Login';

import Upload from './Upload';

import Category from './Category';

import "./App.css";

const App = () => {

const [isLoggedIn, setIsLoggedIn] = useState(false);

useEffect(() => {

const token = localStorage.getItem('fb\_access\_token');

setIsLoggedIn(!!token); // Set `true` if token exists, otherwise `false`

}, []);

const handleLogout = () => {

localStorage.removeItem('fb\_access\_token');

setIsLoggedIn(false); // Update the logged-in state

};

return (

<Router>

{isLoggedIn && <Navbar onLogout={handleLogout} />}

<Routes>

<Route path="/" element={<Login onLogin={() => setIsLoggedIn(true)} />} />

<Route path="/upload" element={<Upload />} />

<Route path="/family" element={<Category category="Family" />} />

<Route path="/friends" element={<Category category="Friends" />} />

<Route path="/places-visited" element={<Category category="Places Visited" />} />

<Route path="/about-me" element={<Category category="About Me" />} />

</Routes>

</Router>

);

};

export default App;

**Login.jsx:**

import React, { useEffect, useState } from 'react';

import { useNavigate } from 'react-router-dom';

import "./App.css";

const Login = ({ onLogin }) => {

const [sdkLoaded, setSdkLoaded] = useState(false);

const [accessToken, setAccessToken] = useState(null);

const [userName, setUserName] = useState('');

const navigate = useNavigate();

useEffect(() => {

window.fbAsyncInit = function () {

window.FB.init({

appId: '1307986103553292', // Replace with your actual App ID

cookie: true,

xfbml: true,

version: 'v16.0', // Latest valid API version

});

setSdkLoaded(true);

console.log('Facebook SDK Initialized');

const token = localStorage.getItem('fb\_access\_token');

if (token) {

validateAccessToken(token);

}

};

(function (d, s, id) {

let js, fjs = d.getElementsByTagName(s)[0];

if (d.getElementById(id)) return;

js = d.createElement(s);

js.id = id;

js.src = '<https://connect.facebook.net/en_US/sdk.js>';

fjs.parentNode.insertBefore(js, fjs);

})(document, 'script', 'facebook-jssdk');

}, [navigate]);

const validateAccessToken = (token) => {

window.FB.api('/me', { access\_token: token }, (response) => {

if (response && !response.error) {

setAccessToken(token);

fetchUserInfo();

onLogin(); // Notify parent component of login

navigate('/upload');

} else {

console.error('Invalid or expired access token. Logging out...');

localStorage.removeItem('fb\_access\_token');

setAccessToken(null);

}

});

};

const handleLogin = () => {

if (!sdkLoaded) {

console.error('Facebook SDK not loaded yet');

return;

}

window.FB.getLoginStatus((response) => {

if (response.status === 'connected') {

const token = response.authResponse.accessToken;

setAccessToken(token);

localStorage.setItem('fb\_access\_token', token);

fetchUserInfo();

onLogin(); // Notify parent component of login

navigate('/upload');

} else {

window.FB.login(

(response) => {

if (response.authResponse) {

const token = response.authResponse.accessToken;

setAccessToken(token);

localStorage.setItem('fb\_access\_token', token);

fetchUserInfo();

onLogin(); // Notify parent component of login

navigate('/upload');

} else {

console.error('User cancelled login or did not fully authorize');

}

},

{ scope: 'public\_profile,email,user\_posts' }

);

}

});

};

const fetchUserInfo = () => {

window.FB.api('/me', { fields: 'name' }, (response) => {

if (response && !response.error) {

setUserName(response.name);

} else {

console.error('Failed to fetch user information:', response.error);

}

});

};

return (

<div className="loginpage">

<h1>Login with Facebook</h1>

{!accessToken ? (

<div className="but">

<button onClick={handleLogin}>Login with Facebook</button></div>

) : (

<div>

<h2>Welcome, {userName}!</h2>

</div>

)}

</div>

);

};

export default Login;

**Navbar.jsx:**

import React from 'react';

import { Link, useNavigate } from 'react-router-dom';

import "./App.css";

const Navbar = ({ onLogout }) => {

const navigate = useNavigate();

const handleLogoutClick = () => {

onLogout();

navigate('/');

};

return (

<nav>

<Link to="/upload">Home</Link>

<Link to="/family">Family</Link>

<Link to="/friends">Friends</Link>

<Link to="/places-visited">Places Visited</Link>

<Link to="/about-me">About Me</Link>

<button

onClick={handleLogoutClick}

>

Logout

</button>

</nav>

);

};

export default Navbar;

**Category.jsx:**

import React, { useEffect, useState } from 'react';

import "./App.css";

const Category = ({ category }) => {

const [posts, setPosts] = useState([]);

const [filteredPosts, setFilteredPosts] = useState([]);

const [errorMessage, setErrorMessage] = useState('');

useEffect(() => {

const accessToken = localStorage.getItem('fb\_access\_token');

if (!accessToken) {

setErrorMessage('Access token not found. Please log in again.');

return;

}

// Fetch user's posts

window.FB.api(

'/me/posts',

'GET',

{ access\_token: accessToken, fields: 'message,created\_time,full\_picture' },

(response) => {

if (response && !response.error) {

setPosts(response.data || []);

} else {

setErrorMessage(

response.error ? response.error.message : 'Error fetching posts.'

);

}

}

);

}, []);

useEffect(() => {

// Filter posts based on the category

const filterPosts = () => {

switch (category) {

case 'Family':

return posts.filter(post =>

post.message?.toLowerCase().includes('family')

);

case 'Friends':

return posts.filter(post =>

post.message?.toLowerCase().includes('friend')

);

case 'Places Visited':

return posts.filter(post =>

post.message?.toLowerCase().includes('visited') ||

post.message?.toLowerCase().includes('travel') ||

post.message?.toLowerCase().includes('trip') ||

post.message?.toLowerCase().includes('visit')

);

case 'About Me':

return posts.filter(post =>

post.message?.toLowerCase().includes('about me') ||

post.message?.toLowerCase().includes('bio') ||

post.message?.toLowerCase().includes('i am') ||

post.message?.toLowerCase().includes('me')

);

default:

return posts;

}

};

setFilteredPosts(filterPosts());

}, [category, posts]);

return (

<div className="mainpage">

<h1>{category} Posts</h1>

{errorMessage && <p style={{ color: 'red' }}>{errorMessage}</p>}

{!errorMessage && filteredPosts.length === 0 && (

<p>No posts available in this category.</p>

)}

<div style={{ display: 'flex', flexWrap: 'wrap', gap: '16px' }}>

{filteredPosts.map((post, index) => (

<div

key={index}

style={{

border: '1px solid #ccc',

padding: '16px',

borderRadius: '8px',

width: '300px',

}}

>

<p>

<strong>Message:</strong> {post.message || 'No message'}

</p>

<p>

<strong>Created:</strong> <span className="layout" >{new Date(post.created\_time).toLocaleString()}</span>

</p>

{post.full\_picture && (

<img

src={post.full\_picture}

alt="Post"

style={{ width: '100%', borderRadius: '8px' }}

/>

)}

</div>

))}

</div>

</div>

);

};

export default Category;

**Upload.jsx:**

import React, { useEffect, useState } from 'react';

const Upload = () => {

const [posts, setPosts] = useState([]);

const [errorMessage, setErrorMessage] = useState('');

useEffect(() => {

const accessToken = localStorage.getItem('fb\_access\_token');

if (!accessToken) {

setErrorMessage('Access token not found. Please log in again.');

return;

}

// Fetch user's posts

window.FB.api(

'/me/posts',

'GET',

{ access\_token: accessToken, fields: 'message,created\_time,full\_picture' },

(response) => {

if (response && !response.error) {

setPosts(response.data || []);

} else {

setErrorMessage(

response.error ? response.error.message : 'Error fetching posts.'

);

}

}

);

}, []);

return (

<div className="mainpage">

<h1>Your Facebook Posts</h1>

{errorMessage && <p style={{ color: 'red' }}>{errorMessage}</p>}

{!errorMessage && posts.length === 0 && <p>No posts available to display.</p>}

<div style={{ display: 'flex', flexWrap: 'wrap', gap: '16px' }}>

{posts.map((post, index) => (

<div

key={index}

style={{

border: '1px solid #ccc',

padding: '16px',

borderRadius: '8px',

width: '300px',

boxShadow:'0px 1px 1px black'

}}

>

<p>

<strong>Message:</strong> {post.message || 'No message'}

</p>

<p>

<strong>Created:</strong> <span className="layout">{new Date(post.created\_time).toLocaleString()}</span>

</p>

{post.full\_picture && (

<img

src={post.full\_picture}

alt="Post"

style={{ width: '100%', borderRadius: '8px' }}

/>

)}

</div>

))}

</div>

</div>

);

};

export default Upload;