**Modernizing M365 Web Application: Objectives and Achievements**

## **1. Introduction**

The modernization of the M365 web application was undertaken to improve performance, enhance responsiveness, and integrate modern development practices. The new implementation leverages **React 18** and several upgraded libraries, enabling better user experience, improved maintainability, and enhanced security.

## **2. Why We Modernized**

The primary objectives of modernizing the M365 web application include:

### **a) Enhanced Performance**

* Implementation of **concurrent rendering** in React 18 for smoother interactions and reduced UI blocking.
* Optimized API handling with an updated **Redux Toolkit** and **Axios** version.
* Reduced page load time and improved responsiveness for better user engagement.

### **b) Improved Responsiveness**

* Designed a fully **mobile-optimized** and adaptive UI.
* Introduced **dynamic popups and deep linking** for seamless navigation.
* Enhanced smooth scrolling and better touch interaction support.

### **c) Better State Management & Data Handling**

* Adoption of **Redux Toolkit** for improved state management, reducing unnecessary re-renders.
* Efficient data flow and handling with optimized API calls and caching mechanisms.

### **d) Modernized UI and UX**

* Integration of **Fluent UI React Components v9** for a polished and accessible interface.
* Improved **UI consistency** and **better theme adaptability** across platforms.
* Enhanced usability with improved animations and transitions.

### **e) Improved Code Maintainability & Modularity**

* A structured and modular codebase promoting **component reusability**.
* Better **code readability** and maintainability for future enhancements.
* Adoption of modern JavaScript and TypeScript best practices.

### **f) Security and Compliance Improvements**

* Upgraded to the latest **Microsoft Teams SDK and SPFx controls** for compliance.
* Strengthened security measures, ensuring better data protection.
* Regular dependency updates to mitigate vulnerabilities.

## **3. What We Achieved**

The modernization of the M365 web application resulted in significant improvements:

| **Feature** | **Old M365 (React 16)** | **New M365 (React 18)** |
| --- | --- | --- |
| **Performance** | Slower UI updates due to synchronous rendering | Faster UI with concurrent rendering |
| **State Management** | Basic Redux, leading to performance issues | Redux Toolkit for optimized data handling |
| **UI Framework** | Old Fluent UI components | Fluent UI v9 with better styling & UX |
| **Responsiveness** | Static layouts, poor mobile support | Fully responsive, mobile-optimized |
| **Deep Linking** | Limited functionality | Full deep linking for better navigation |
| **Popup Management** | Basic popups | Dynamic popups with mobile-friendly behavior |
| **Security Compliance** | Outdated Teams SDK, security risks | Updated SDK ensuring compliance |
| **Routing** | React Router v5 (older version) | React Router v6 for better navigation |
| **API Calls & Optimization** | Older Axios version, potential memory leaks | Updated Axios & better API error handling |
| **Polyfills & Smooth Scroll** | Limited browser support | Enhanced smooth scrolling & polyfills |

## **4. Conclusion**

The modernization of the M365 web application to **React 18** and the latest libraries has significantly improved its performance, scalability, and usability. This transition ensures that the application remains future-proof, providing users with a faster, more secure, and seamless experience across devices. Moving forward, this modernized infrastructure will allow easier integration with new Microsoft 365 features and APIs.