

prog7312

Technology Recommendations



November 17, 2024

VCWaterfall

ST10091991

1. **Cloud Storage and Synchronization (e.g., Microsoft Azure Blob Storage or Amazon S3)  
   Recommendation**:   
   By offering scalable, secure, and dependable storage for substantial volumes of user-generated data, including service request reports, user-uploaded files, and photos, cloud storage solutions like Amazon S3 and Microsoft Azure Blob Storage can improve the application.

**Justification**:  
**Scalability**: Cloud storage guarantees that the application can grow without necessitating major infrastructure adjustments as the volume of service requests and user interactions rises.  
**Reliability**: Cloud services guarantee data protection and accessibility even in the event of system failures by providing high availability and redundancy.  
**Smooth Integration**: These storage options work well with web and mobile apps, particularly when used in cloud-native architectures. For example, Azure Blob Storage is simple to integrate with MAUI apps, guaranteeing dependability and consistency.

1. **Push Notifications (e.g., Firebase Cloud Messaging (FCM))**  
   **Recommendation**:   
   The application will notify users in real time about status updates, new service requests, or any problems that need their attention by implementing push notifications with Firebase Cloud Messaging (FCM).

**Justification**:  
**Real-Time Updates**: Without opening the program, customers can receive push alerts to stay up to speed on the status of their service requests and any new developments.  
**User Engagement**: By providing non-intrusive information, push notifications aid in improving user engagement and retention.  
**Cost-Effective**: FCM offers a reasonably priced notification delivery system that is simple to incorporate into MAUI apps. Web applications, iOS, and Android are among the platforms it supports.

1. **GraphQL for Efficient Data Fetching**  
   **Recommendation**:  
   Performance might be enhanced by using GraphQL for data fetching rather than REST since it would enable the application to request only the data it requires, saving bandwidth and speeding up load times.

**Justification**:  
**Optimised Data Queries**: GraphQL minimises over-fetching of superfluous data by letting clients specify precisely what data they require. Because of this, it is perfect for mobile applications whose performance depends on transferring as little data as possible.  
**Single Endpoint**: GraphQL employs a single endpoint, which streamlines the backend architecture and enhances maintainability in contrast to REST, which may call for several endpoints for various data kinds.  
**Efficiency**: GraphQL can improve overall performance and response times by retrieving only the necessary data, particularly on sluggish networks or when working with big databases.  
**Compatibility**: GraphQL works with a number of backend technologies, such as Node.js,.NET, and Python, and may be integrated with MAUI apps through API queries.

1. **Continuous Integration/Continuous Deployment (CI/CD) Tools (e.g., GitHub Actions, Azure DevOps, or CircleCI)**  
   **Recommendation**:  
   Development and deployment procedures would be streamlined by putting CI/CD solutions like CircleCI, Azure DevOps, and GitHub Actions into use. These solutions guarantee quicker delivery and fewer bugs in production by automatically building, testing, and deploying the application anytime changes are made to the codebase.

**Justification**:  
**Automated Builds and Tests**: CI/CD pipelines make sure that code modifications are tested automatically, which lowers the possibility of introducing defects and speeds up development.  
**Consistent Deployment**: CI/CD enables developers to swiftly and reliably release updates to production, minimising downtime and enhancing user experience.  
**Efficiency of Collaboration**: By eliminating human labour, automated procedures enhance developer collaboration, freeing up developers to concentrate more on feature creation and less on deployment and testing.  
**Compatibility**: MAUI projects offer optimised pipelines for development, testing, and deployment through easy integration with well-known CI/CD systems.

**Conclusion**  
The Municipal Services Application's efficiency and usefulness might be greatly improved by integrating these further technologies. These technologies offer significant benefits that complement the project's objectives, whether through enhanced data management, a smooth user experience, or predictive capabilities. These suggestions are in line with the application's present architecture and may provide noticeable enhancements to the system's overall usability, efficiency, and scalability.