# Enhancing VisionMate with Progressive Web App (PWA)

#### Sai Anoushka K CS 298 Dr. Chris Pollett

## Intro to VisionMate

What is VisionMate?

- Al-powered image captioning web app.
- Uses React (frontend), FastAPI (backend), Hugging Face Spaces (model deployment).

Current Status: Web app with an interactive UI for generating image descriptions



# Progressive Web App (PWA)

#### What is PWA ?

# **Definition:** A PWA is a web app with native app-like capabilities.

#### **Core Features:**

- Reliable Works offline using caching.
- **Fast** Loads quickly and provides a smooth experience.
- Engaging Installable on devices, just like a native app.

### Why PWA for our App?

- Faster loading times for image processing.
- Offline mode: Users can still access past captions.
- No app store needed: No installation friction.
- **Push notifications:** Reminders and updates.

## PWA vs Traditional Web App

Feature	PWA	Traditional Web App
Offline Support		×
Needs Domain Name		
Installable	🔽(via Web )	×
Push Notifications		×
Performance	🔽 (Faster)	🗙 (Slower)



# Key PWA Technologies

- Service Workers Enables caching & offline mode.
- Web App Manifest Makes the app installable.
- **HTTPS** Ensures security.
- Lazy Loading & Caching Improves speed.



## Steps to convert VisionMate to a PWA

- Add a Web App Manifest
- Implement Service Workers for offline support
- Enable HTTPS for security
- Optimize performance (lazy loading, caching)



#### **Challenges & Considerations**

#### **Conclusion & Next Steps**

- Some iOS limitations (limited offline support)
- Requires additional development
- Background processes not as powerful as native apps

PWA is a great fit for VisionMate! 🚀

If approved, next steps:

- Implement PWA features in React.
- Test offline functionality & performance.
- Deploy enhanced VisionMate as a PWA.