Re: Technology Stack for Project Pragati

---

1. Core Languages

- **TypeScript:** This will be our primary language. It is a superset of JavaScript that adds strong typing. For a large, complex application like this (especially with accounting and finance modules), TypeScript is essential for ensuring code quality, scalability, and long-term maintainability.
- **JavaScript (ES6+):** The foundational language for the entire stack.

---

2. Backend (Server-side)

This is the central "brain" of the application that will power the web, mobile, and desktop clients.

- **Node.js:** The runtime environment we'll use to run the server-side code.
- **Express.js:** A fast and minimal framework for building the core **RESTful API** that all applications will communicate with.
- **WebSockets (e.g., Socket.io):** This is critical for the "Real-time Data Sync" feature mentioned in Module 8.

---

3. Frontend & Client Applications

This covers all the user-facing applications.

- **Web Application: React.js**. A powerful industry-standard library for building a fast, modern, and responsive user interface for the web.
- **Mobile Application: React Native**. A framework that allows us to use one codebase (in React) to build a truly native application for both **iOS and Android**. This is crucial for features like native camera scanning (Module 3) and push notifications (Module 8).
- **Desktop Application: Electron.js**. This framework allows us to package the web application into a high-performance, installable desktop app that works on **Windows, macOS, and Linux**.

---

4. Database

- **PostgreSQL:** This will be our primary and only database. It is a powerful, open-source, and highly reliable SQL database. It is perfectly suited for handling complex financial transactions, ensuring data integrity (a must for an accounting app), and scaling as your user base grows.

---

5. Cloud Infrastructure

- **Microsoft Azure:** We will host the entire application backend, database, and all related services on Azure. This provides a secure, scalable, and reliable foundation.
  - **Azure App Service:** To host the Express.js backend API.
  - **Azure Database for PostgreSQL:** A fully managed service to run our PostgreSQL database, handling backups, security, and scaling.
  - **Azure Blob Storage:** To securely store all file uploads, such as scanned purchase bills, company logos, and reports.
  - **Azure Functions:** For running serverless tasks like automatic payment reminders.

---

6. Key Integrations & Services

Based on the advanced features in your proposal, the core stack will also be integrated with several key third-party services:

- **Authentication: JSON Web Tokens (JWT)** for securing the API (Module 1).
- **GST & Accounting:** Integration with **GSTN APIs** (likely via a GSP) for GSTR reporting (Module 6) and handling **Tally XML** formats for import/export (Module 8).
- **Marketing & Comms: WhatsApp Business API** (Module 7) and **Firebase Cloud Messaging (FCM)/Apple Push Notification Service (APNS)** for payment reminders (Module 8).
- **AI & Data:** An **OCR Service** (like Azure Cognitive Services) for the "Scan/Upload Purchase Bills" feature (Module 8).
- **Finance:** Integration with an **Account Aggregator (AA) platform** (like Setu or Finvu) to enable the "Online Bank Account Linking" feature (Module 4).

This technology stack is modern, scalable, and perfectly suited to deliver on all the features outlined in the proposal.