**📋 JV Web App - Project Context Document**

**🎯 Project Overview**

**Purpose: Build a comprehensive web interface for live Joint Venture (JV) negotiation and MoU generation, operated by Partner C during meetings.**

**Core Business Model:**

* **Partners: A (Plant Owner), B (Market/Dealer), C (Operations Expert)**
* **Equity Split: A=40%, B=42%, C=18%**
* **Key Value: Trailer manufacturing with ₹50K margin/unit**

**🏗️ Architecture Decisions**

**1. Technology Stack**

* **Frontend: Vanilla HTML/CSS/JavaScript (no framework)**
* **State Management: Custom lightweight store (jvStore.js)**
* **Charts: Chart.js**
* **Excel Export: XLSX library**
* **Styling: CSS variables with dark theme**
* **No Backend: All calculations client-side**

**2. File Structure**

**jv-web-app/**

**├── index.html # Main app shell**

**├── styles/**

**│ └── main.css # Global styles**

**├── scripts/**

**│ ├── main.js # Navigation & initialization**

**│ └── jvStore.js # Centralized state (TO BE CREATED)**

**├── modules/**

**│ └── calculator/ # Working Capital Simulator**

**│ ├── index.html**

**│ ├── css/style.css**

**│ ├── js/script.js**

**│ └── components/**

**│ ├── balance-sheet.html**

**│ ├── cashflow-form.html**

**│ └── pnl-statement.html**

**└── project\_log.md # Development history**

**📊 Data Flow Architecture**

**Centralized Store Structure (Planned)**

**JVStore = {**

**state: {**

**overview: { partners, timeline },**

**financialModel: { workingCapital, cashFlow, pnl, balanceSheet },**

**contributions: { values, equitySplit },**

**governance: { roles, voting, profitPolicy },**

**// ... other sections**

**}**

**}**

**Key Data Dependencies**

1. **Working Capital Simulator → Determines Partner B investment**
2. **Partner B Investment → Affects contribution totals**
3. **Contribution Totals → Determines equity split**
4. **Equity Split → Affects profit distribution**
5. **All Sections → Feed into MoU generation**

**✅ Completed Features**

**1. Working Capital Simulator (Section 2)**

* **✅ Cash flow simulation with GST and payment delay options**
* **✅ Automatic Partner B investment calculation**
* **✅ P&L statement generation**
* **✅ Balance sheet creation**
* **✅ Excel export functionality**
* **✅ Interactive charts**

**2. UI Framework**

* **✅ Collapsible sidebar navigation**
* **✅ 11 section containers**
* **✅ Dark theme with accent colors**
* **✅ Section 1 & 3 basic UI (no logic yet)**

**🚧 Implementation Status**

| **Section** | **Name** | **UI Status** | **Logic Status** | **Store Integration** |
| --- | --- | --- | --- | --- |
| **1** | **Overview & Setup** | **✅ Complete** | **⚠️ LocalStorage only** | **❌ Pending** |
| **2** | **Working Capital Simulator** | **✅ Complete** | **✅ Complete** | **❌ Pending** |
| **3** | **Contribution & Equity** | **✅ Basic UI** | **❌ No calculations** | **❌ Pending** |
| **4** | **Roles & Responsibilities** | **❌ Placeholder** | **❌ Not started** | **❌ Pending** |
| **5** | **Profit & Reserve Policy** | **❌ Placeholder** | **❌ Not started** | **❌ Pending** |
| **6** | **Governance Matrix** | **❌ Placeholder** | **❌ Not started** | **❌ Pending** |
| **7** | **Lock-in & Exit** | **❌ Placeholder** | **❌ Not started** | **❌ Pending** |
| **8** | **Conflict Resolution** | **❌ Placeholder** | **❌ Not started** | **❌ Pending** |
| **9** | **Scenario Simulator** | **❌ Placeholder** | **❌ Not started** | **❌ Pending** |
| **10** | **MoU Drafting** | **❌ Placeholder** | **❌ Not started** | **❌ Pending** |
| **11** | **Config & Archive** | **❌ Placeholder** | **❌ Not started** | **❌ Pending** |

**🔧 Technical Decisions & Patterns**

**1. Component Loading Pattern**

**// Dynamic HTML component loading**

**async function loadComponent(componentPath, targetId) {**

**const response = await fetch(componentPath);**

**const html = await response.text();**

**document.getElementById(targetId).innerHTML = html;**

**}**

**2. Event Handling Pattern**

* **Use event delegation where possible**
* **Store references in variables to avoid repeated DOM queries**
* **Namespace events by section**

**3. State Update Pattern (Planned)**

**// Always update store, never DOM directly**

**JVStore.setState({ section: { field: value } });**

**// Store notifies subscribers**

**// Subscribers update DOM**

**📝 Critical Business Rules from Partnership Model**

**Core Financial Parameters**

1. **Partner Contributions (Conservative Values):**
   * **Partner A: ₹2 Cr (existing plant under A's entity, used via formal lease)**
   * **Partner BC Combined: ₹3 Cr total** 
     + **Working Capital: ₹2 Cr (from simulator)**
     + **Plant Renovation: ₹50L**
     + **Market Value (B): ₹40L**
     + **Operations Value (C): ₹10L**
   * **Total Pool: ₹5 Cr → A=40%, BC=60% (B=42%, C=18%)**
2. **Operational Economics:**
   * **Unit Cost: ₹13.5L**
   * **Unit Sale Price: ₹14L**
   * **Margin per Unit: ₹50,000**
   * **Production Cycle: 7 days → 6 units**
   * **Quarterly Target: 13 cycles = 78 units = ₹39L margin**
3. **Salary Structure (Treated as OpEx):**
   * **Partner A: ₹12 LPA (₹3L/quarter)**
   * **Partner C: ₹18 LPA (₹4.5L/quarter)**
   * **Partner B: No salary mentioned**

**Profit Distribution Rules**

1. **Calculation: Gross Profit - Salaries - Running Costs = Net Profit**
2. **Timing: Calculated at each quarter-end**
3. **Disbursement:** 
   * **50% of profit disbursed at next quarter-end**
   * **Partner A gets monthly draws against profit (due to EMIs)**
   * **40% retained for reinvestment (capped at ₹1 Cr)**

**Governance Framework**

1. **Tier 1 - Operational (Domain Authority):**
   * **Finance & Purchase: Partner A (veto right)**
   * **Market & Orders: Partner B (final say)**
   * **Plant Operations: Partner C (daily autonomy)**
2. **Tier 2 - Strategic (≥70% vote required):**
   * **CapEx expansion, major hires, long-term contracts**
   * **Can be A+B or B+C**
3. **Tier 3 - Extraordinary (Unanimous/≥75%):**
   * **Ownership changes, JV dissolution**

**Lock-in & Exit Rules**

1. **Lock-in Periods: A=3 years, B&C=2 years**
2. **Breach Penalties:** 
   * **Partner A: Forfeits 15% of equity value**
   * **Partner B: Forfeits ₹50L or equivalent %**
   * **Partner C: Forfeits entire sweat equity**
3. **Mediation: 15-day grace period before penalties**
4. **Valuation Method: Book value + 3× average EBITDA (last 4 quarters)**

**🎯 Next Implementation Steps**

1. **Create jvStore.js with state management system**
2. **Integrate Working Capital Simulator with store**
3. **Complete Section 3 logic (Contribution & Equity calculations)**
4. **Build Section 4 (Roles & Responsibilities)**
5. **Create reusable UI components (tables, forms, cards)**

**💡 Development Guidelines**

**Before Each Session:**

* **Share this document**
* **Share latest project\_log.md**
* **Specify which section/feature to work on**

**Code Patterns to Follow:**

* **Use semantic HTML with proper ARIA labels**
* **Keep styles modular by section**
* **Comment complex calculations**
* **Use consistent naming: camelCase for JS, kebab-case for CSS**

**Testing Checklist:**

* **[ ] Calculations match manual verification**
* **[ ] State persists across page reloads**
* **[ ] All inputs validate properly**
* **[ ] Excel exports contain correct data**

**🚨 Known Issues & Constraints**

1. **No localStorage in Section 2 - Calculator data not persisted**
2. **Manual sync needed - Sections don't auto-update each other yet**
3. **No validation - Input fields accept any values**
4. **Fixed GST/delay logic - Hardcoded in calculator, needs to be configurable**

**📚 Reference Information**

* **Calculator Logic: See modules/calculator/js/script.js for complex calculations**
* **Business Model: See attached partnership model.docx for detailed rules**
* **UI Specs: See JV web interface layout.docx for component details**

**Required Reference Documents**

1. **partnership model.docx - Contains all business rules, contribution values, governance framework, and legal terms. This is the source of truth for all calculations and validations.**
2. **JV web interface layout.docx - UI/UX specifications for each section**

**How to Use This Document**

1. **Start each new workspace by sharing:**
   * **This context document**
   * **Latest project\_log.md**
   * **Current git repository state**
2. **Specify your goal: "Continue from [last log entry]. Today I want to [specific task]"**
3. **Update after each session:**
   * **Add entry to project\_log.md**
   * **Update this document if architecture changes**

**This approach ensures continuity regardless of which ChatGPT instance you're using. The combination of this context document + project\_log.md + git repository gives complete project state.**