Frontech Innovations

GSTIN: **07BMXPL6116Q1Z3**

Ready to develop new thing

Welcome to Frontech Innovations!

We are thrilled to work with you At Frontech Innovations, we are committed to delivering innovative solutions tailored to your needs

Quotation for Software Development

- **Content management system**

Tech Stack Used

1. **Backend**
   - **Node.js**: Used as the server-side runtime to manage APIs and real-time communication.
   - **Express.js**: Framework to handle HTTP requests, routing, and middleware.
   - **Socket.IO**: Enables real-time, two-way communication between the main server and client displays.
   - **Multer**: Handles file uploads (PDFs, videos, checklists) from the main computer.
   - **File System (fs)**: Manages local file storage for uploaded content.
   - **SQLite**: Lightweight database to store metadata about files and content assignments.
2. **Frontend**
   - **HTML5**: Provides the structure for the user interface.
   - **CSS3**: Styles the admin dashboard and client displays.
   - **Vanilla JavaScript**: Adds interactivity and dynamic behavior to the frontend.
   - **Optional CSS Framework**: Tailwind CSS or Bootstrap for responsive and clean design (if required).
3. **Networking**
   - Local network using a static IP address for the main computer to serve client devices.
4. **Tools**
   - **Postman**: To test API endpoints.
   - **VS Code**: For code development.
   - **PM2**: To run the Node.js server as a background service on the main computer.

How the Project Will Work

1. **Main Computer (Server):**
   - Hosts a local Node.js server that manages content uploads and distribution.
   - Admins can log in via a web-based dashboard to upload or update content (PDFs, videos, checklists) and assign it to specific tabs.
   - The server stores uploaded content in a structured folder system and tracks metadata in an SQLite database.
   - Real-time updates are sent to client displays using Socket.IO whenever content changes.
2. **Client Displays (Read-Only Mode):**
   - Each client display connects to the Node.js server via a local IP address.
   - The client fetches assigned content based on the tab it is configured to display.
   - Content is displayed dynamically using a simple HTML and CSS interface.
   - Real-time updates from the server ensure that displays reflect the latest content without manual refreshes.
3. **Networking and Deployment:**
   - All devices (main computer and clients) are connected to the same local network via LAN or Wi-Fi.
   - The main computer is assigned a static IP to ensure consistent access by client devices.
   - The Node.js server runs continuously on the main computer, with client displays accessing it through a browser or lightweight app.

Example Workflow:

1. The admin logs into the main computer’s dashboard and uploads a video for Tab 3.
2. The server stores the video locally and updates the database.
3. The server notifies all connected clients about the update for Tab 3 using Socket.IO.
4. The client assigned to Tab 3 fetches and displays the updated video immediately.

This system ensures efficient content management and display, making it ideal for manufacturing environments that require precise, real-time information dissemination.

Payments Quotations

Full Phase:	30,000
Advance:	5,000

Total + GST : 30,000 + 5,400 = 35,400

Estimated time : 15 Days approx..

Terms and Conditions

Payment Terms:	Remaining payments will be determined based on the completion of different phases of development
Quotation Validity:	- This quotation is valid for 10 days from the date of issuance - Prices and terms are subject to change after this period

Note:-

- Additional terms and conditions may apply, please refer to the detailed agreement for full information.

- For any queries or clarifications regarding the terms, please contact us.