3_FRONTEND_DOCUMENTATION —

Project structure, key files explained, how OpenVidu Browser SDK is used, setup and run locally, how to build and deploy to production, update deployed frontend after changes.

Frontend Documentation

ChourangiHealth Video Consultation Platform

Technology Stack

- React.js 19
- OpenVidu Browser SDK 2.32.1
- jwt-decode 4.0.0
- Create React App 5.0.1

Project Structure

```
openvidu-frontend-poc/
│
├── src/
│   ├── App.js            # Root component, handles login state
│   ├── index.js          # React entry point
│   ├── App.css           # Global styles
│   │
│   ├── api/
│   │   └── api.js        # API calls to backend (login, video session)
│   │
│   └── pages/
│       ├── LoginPage.js     # Login page with role selection
│       └── VideoPage.js     # Video consultation page
│
├── public/               # Static assets
├── .gitignore
├── package.json
└── package-lock.json
```

Key Files Explained

src/api/api.js

Contains two functions:

loginUser — Sends POST request to backend login endpoint. On success decodes the JWT token using jwt-decode to extract the role and stores both the token and role in localStorage.

createVideoSession — Sends POST request to backend video session endpoint with the Calling ID and JWT token in the Authorization header. Returns session data including the OpenVidu token.

The API base URL is set to point to the production server:

```
const API_BASE_URL = "https://chourangi.duckdns.org";
```

Update this value when changing domains or running locally.

src/pages/LoginPage.js

The login page features:

- Soft purple and white theme with floating blob background effects
- Two panel layout with left panel showing illustrations and right panel showing the login form
- Cartoon SVG illustrations of doctor and patient facing each other
- Role toggle between Doctor and Patient which auto fills username
- Username and password input fields
- Error display for invalid credentials
- Purple gradient Sign In button
- Demo credentials hint at the bottom

When the Doctor role is selected username is auto filled with doctor. When Patient is selected username is auto filled with user. Password is always 123456 for this POC.

src/pages/VideoPage.js

The video page features:

- Frosted glass header with brand name, call timer, role badge, connection status, control buttons
- Join screen with cartoon avatar, Calling ID input, and Start Consultation button
- Large remote video box with waiting animation (bouncing dots and avatar) while waiting for the other participant
- Small picture-in-picture local video in the bottom right corner
- Video on/off toggle button that shows red when video is on and green when off
- Audio mute/unmute toggle button with same color logic
- Leave Call button
- Live call duration timer that starts when both participants are connected

Key technical implementation details:

The streamCreated event uses a 300 millisecond setTimeout before calling subscriber.addVideoElement to ensure the React DOM has rendered the video element:

```
mySession.on("streamCreated", (event) => {
   const subscriber = mySession.subscribe(event.stream, undefined);
   setRemoteJoined(true);
   setTimeout(() => {
      if (remoteVideoRef.current) {
         subscriber.addVideoElement(remoteVideoRef.current);
      }
   }, 300);
});
```

The publisher is also initialized after a 300 millisecond delay after connecting to the session for the same reason.

Session disconnection is handled gracefully. If the disconnection reason is networkDisconnect or reconnectTimeout the app automatically attempts to rejoin after 3 seconds.

Setup and Run Locally After Git Pull

Step 1 — Clone the repository

```
git clone -b master https://github.com/AshuuuPatil/openvidu_videocall_microservice_frontend.git
frontend
cd frontend
```

Step 2 — Install dependencies

```
npm install
```

Step 3 — Update API URL for local development

Open src/api/api.js and change the API base URL:

```
// For local development:
const API_BASE_URL = "http://localhost:8000";

// For production:
const API_BASE_URL = "https://chourangi.duckdns.org";
```

Step 4 — Run the development server

```
npm start
```

The app opens automatically at http://localhost:3000

Build for Production

```
npm run build
```

This creates a build folder with optimized production files. Copy these files to the server:

```
cp -r build/* /opt/openvidu/custom-layout/
```

Then restart OpenVidu to serve the new build.

Update Deployed Frontend After Code Changes

After making changes locally and pushing to GitHub, on the EC2 server:

```
cd /home/ubuntu/test-1/frontend
git pull origin master
npm run build
```

```
cp -r build/* /opt/openvidu/custom-layout/
```

No restart needed — Nginx serves the files directly and they are updated immediately.

How OpenVidu Browser SDK is Used

The frontend uses OpenVidu Browser SDK version 2.32.1. The flow is:

1. Initialize OpenVidu: const OV = new OpenVidu()
2. Initialize session: const mySession = OV.initSession()
3. Register event handlers for streamCreated, streamDestroyed, sessionDisconnected
4. Connect to session using the token received from backend: await mySession.connect(token)
5. Initialize publisher with camera and microphone: await OV.initPublisherAsync(undefined, options)
6. Publish stream: await mySession.publish(publisher)
7. Attach local video to HTML element: publisher.addVideoElement(localVideoRef.current)
8. When remote stream arrives attach it: subscriber.addVideoElement(remoteVideoRef.current)

---

localStorage Keys Used

Key	Value	Purpose
jwt	JWT token string	Used in Authorization header for API calls
role	admin or user	Determines if current user is doctor or patient

Both keys are removed on logout.

---

npm Scripts

```
npm start       # Start development server on localhost:3000
npm run build   # Build production files to build/ folder
npm test        # Run tests
npm run eject   # Eject from Create React App (irreversible)
```

Dependencies

Main dependencies from package.json:

```
{
  "openvidu-browser": "^2.32.1",
  "jwt-decode": "^4.0.0",
  "react": "^19.2.4",
  "react-dom": "^19.2.4",
  "react-scripts": "5.0.1"
}
```