Project Setup Instructions – ClinicPro

Backend Setup

1. Install Dependencies

Run the following command to install all required packages:

```
npm install
```

---

2. Environment Configuration

Create a .env file in the root directory and add the following variables:

```
PORT=3000
MONGODB_URI=mongodb://localhost:27017/clinic-pro
NODE_ENV=development
JWT_SECRET=your_jwt_secret_here
JWT_EXPIRES_IN=7d
```

Note: Replace your_jwt_secret_here with a secure, random string.

---

3. Database Setup

Ensure MongoDB is running properly on your system.

- Local MongoDB: Start the MongoDB service (e.g., mongod or via services).
- MongoDB Atlas: Replace the MONGODB_URI value in .env with your Atlas connection string.

---

4. Run Database Seeders (Optional)

To populate the database with initial data:

```
npm run seed
```

To clear existing data and reseed:

```
npm run seed:clear
```

---

5. Start the Server

Development Mode:

```
npm run dev
```

Production Mode:

```
npm start
```

---

6. Build for Production

Build the project for deployment:

```
npm run build
```

Frontend Setup

1. Install Dependencies

Run the following command to install all required packages:

```
npm install
```

---

2. Development Server

Start the development server with:

```
npm run dev
```

Once started, the application will be available at:
http://localhost:5173

Environment Configuration – ClinicPro

Backend .env

```
# Application
FRONTEND_URL=http://localhost:3000
PORT=3000
NODE_ENV=development

# Database
MONGODB_URI=mongodb://localhost:27017/clinic-pro

# Authentication
JWT_SECRET=your-secret-key

# AWS S3 Configuration
AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_REGION=
AWS_S3_BUCKET_NAME=

# Optional S3 Settings
AWS_S3_ENDPOINT=https://s3.amazonaws.com
S3_FORCE_PATH_STYLE=false
S3_SIGNED_URL_EXPIRES=3600

# File Upload Settings
UPLOAD_PROVIDER=s3
MAX_FILE_SIZE=52428800
ALLOWED_FILE_TYPES=image/jpeg,image/png,image/gif,image/webp,video/mp4
,video/avi,video/mov,audio/mp3,audio/wav

# S3 Folder Structure (Optional)
S3_AVATARS_FOLDER=avatars
S3_BANNERS_FOLDER=banners
S3_MEDIA_FOLDER=media
S3_THUMBNAILS_FOLDER=thumbnails

# Payment Gateway (Stripe)
STRIPE_SECRET_KEY=
STRIPE_WEBHOOK_SECRET=
```

Frontend .env

```
# API Configuration
VITE_API_BASE_URL=http://localhost:3000/api
NEXT_PUBLIC_API_URL=http://localhost:3000/api

# Application
PORT=3000
```

ClinicPro Deployment on EC2 (with MongoDB Atlas, Node.js, React, PM2, Nginx, Google Domains)

---

1. Connect to EC2 & Install Dependencies

```
# Connect to EC2
ssh -i your-key.pem ubuntu@your-ec2-public-ip

# Update system
sudo apt update && sudo apt upgrade -y

# Install Node.js (LTS)
curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash -
sudo apt-get install -y nodejs

# Verify versions
node -v
npm -v

# Install pm2 globally
sudo npm install -g pm2

# Install build tools (optional but useful)
sudo apt-get install -y build-essential git nginx
```

---

2. Project Structure on EC2

Suppose you cloned your repo into /var/www/clinicpro:

```
/var/www/clinicpro
  ├── backend   (Node.js/Express, TypeScript/TSX build)
  └── frontend  (React / Vite or Next.js)
```

---

3. Configure Backend (Node.js + PM2)

```
cd /var/www/clinicpro/backend

# Install dependencies
npm install

# Build TypeScript
npm run build

# Setup .env file
nano .env
```

Paste your backend .env (with MongoDB Atlas connection string). Example:

```
FRONTEND_URL=https://xyz.com
PORT=3000
NODE_ENV=production
MONGODB_URI=mongodb+srv://user:password@cluster0.mongodb.net/clinic-pro
JWT_SECRET=super-secret-key
```

Start backend with PM2

```
pm2 start dist/index.js --name clinicpro-backend
pm2 save
pm2 startup
```

Backend will now run on port 3000 (or whichever you set in .env).

---

4. Configure Frontend (React Build)

```
cd /var/www/clinicpro/frontend

# Install dependencies
```

```
npm install

# Create .env file
nano .env
```

Example:

```
VITE_API_BASE_URL=https://xyz.com/api
NEXT_PUBLIC_API_URL=https://xyz.com/api
```

Build frontend

```
npm run build
```

This will generate a `dist/` folder (for Vite/React).
We'll serve this with Nginx.

---

5. Setup Nginx Reverse Proxy

```
sudo nano /etc/nginx/sites-available/clinicpro
```

Paste config:

```
server {
    server_name xyz.com www.xyz.com;

    root /var/www/clinicpro/frontend/dist;
    index index.html index.htm;

    location / {
        try_files $uri /index.html;
    }

    location /api/ {
        proxy_pass http://localhost:3000/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Enable site:

```
sudo ln -s /etc/nginx/sites-available/clinicpro /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl restart nginx
```

---

6. Setup SSL with Let's Encrypt

Install Certbot:

```
sudo apt install -y certbot python3-certbot-nginx
sudo certbot --nginx -d xyz.com -d www.xyz.com
```

This will auto-configure HTTPS.

---

7. Domain Setup (Google Domains)

- Go to Google Domains → DNS settings.

Add an A record:

```
 Name: @
Type: A
TTL: 300
Value: <Your EC2 Public IP>
```

-
- Add `www` as a CNAME record pointing to `xyz.com`.

Wait for DNS propagation (~15 mins to a few hours).

---

8. Keep Services Running

- Backend runs with PM2 (clinicpro-backend)
- Frontend is served by Nginx
- SSL auto-renews with Certbot

Check status:

```
pm2 list
systemctl status nginx
```

Final Access

- Backend API → https://xyz.com/api
- Frontend App → https://xyz.com