**Next.js Project Setup**

# Firebase Connection (Inside Server Folder)

1. **Create Firebase Account**: Go to [Firebase](https://firebase.google.com/) and sign up or log in to your account.
2. **Create Project**:
   * Click on "Add project".
   * Follow the steps to create your project (Do not enable Firebase Hosting).
3. **Add Firebase Configuration**:
   * Go to your project settings and copy the Firebase configuration.
   * Create a **utils/firebase.js** file and paste the Firebase configuration into it.

// utils/firebase.js

import firebase from "firebase/app";

import "firebase/auth";

const firebaseConfig = {

apiKey: "your-api-key",

authDomain: "your-auth-domain",

projectId: "your-project-id",

storageBucket: "your-storage-bucket",

messagingSenderId: "your-messaging-sender-id",

appId: "your-app-id"

};

if (!firebase.apps.length) {

firebase.initializeApp(firebaseConfig);

}

export default firebase;

1. **Enable Authentication**:
   * Click on "Develop" in the left sidebar.
   * Go to "Authentication".
   * Click on "Sign-in method" and enable "Google" as a sign-in provider.

# Prisma Connection (Inside Server Folder)

1. **Initialize Prisma**:
   * Run **npx prisma init** to initialize Prisma in your project.
2. **Setup Database**:
   * Go to [Supabase](https://supabase.io/) and create a new project. Alternatively, you can install PostgreSQL locally.
   * In Supabase, go to "Settings".
   * Navigate to "Database" and copy the database connection URL.
   * Paste this URL into your **.env** file as **DATABASE\_URL**.

DATABASE\_URL="your-supabase-database-url"

1. **Push Database Schema**:
   * Run **npx prisma db push** to push the Prisma schema to your database.
2. **View Database Schema**:
   * Run **npx prisma studio** to open Prisma Studio and view your database schema.

# Node Connection (Inside Server Folder)

1. **Start Node Server**:
   * Run **yarn start** to start the Node.js server.

**Example Express Server Setup**

Create an Express server with environment configuration:

1. **Install Dependencies**:
   * Run **yarn add express dotenv cors**.
2. **Create Server File**:
   * Create a **server.js** file with the following content:

import express from "express";

import dotenv from "dotenv";

import cors from "cors";

dotenv.config();

const app = express();

app.use(cors());

app.use(express.json());

const server = app.listen(process.env.PORT, () => {

console.log(`Server started on port ${process.env.PORT}`);

});

**Setup Environment Variables**:

* + Ensure you have a **.env** file with the following content:

PORT=3000

DATABASE\_URL=your\_database\_url

# How to Run Project

* Yarn dev to run project