Business Nexus - Full Stack Development Intern Task

Project Overview:

Business Nexus is a professional networking platform designed to connect entrepreneurs and investors. The objective is to build a full-stack application that allows users to register as either an investor or entrepreneur, view profiles, send collaboration requests, and communicate via a real-time chat system. This document outlines the weekly tasks, tools, and deliverables for full stack development interns.

Tools & Technologies

Frontend:

- **Framework:** React.js (Vite or CRA)
- **Styling:** Tailwind CSS / Styled Components / CSS Modules
- **Routing:** React Router
- **State Management:** Context API / Redux (basic)
- **API Integration:** Axios
- **Version Control:** Git + GitHub
- **UI Design: In this project use your own creativity**

YOU CAN USE TECH STACK OF YOUR OWN CHOICE

Backend:

- **Runtime & Framework:** Node.js + Express.js
- **Database:** MongoDB + Mongoose (or PostgreSQL if required)
- **Authentication:** JWT + bcrypt
- **Real-time:** Socket.io
- **Mail Service (Optional):** Nodemailer
- **Testing:** Postman (API), Jest (Optional)

DevOps & Deployment:

- **Frontend:** Vercel / Netlify
- **Backend:** Render / Railway / Heroku

- **API Testing:** Postman

WEEK 1: Project Setup + Auth System + Basic Layouts

Goals:

- Set up both frontend and backend environments
- Implement authentication APIs and connect with the frontend
- Create basic routes and layouts for both user roles

Tasks:

Frontend:

1. **Project Setup**
   - Create a new React project using Vite or CRA.
   - Organize folders: `components`, `pages`, `layouts`, `assets`, `services`, etc. ○ Install dependencies: React Router, Axios, Tailwind CSS (or preferred styling method).
2. **Routing Setup**
   - Configure the following routes:
     - `/login`
     - `/register`
     - `/dashboard/investor`
     - `/dashboard/entrepreneur`
     - `/profile/investor/:id`
     - `/profile/entrepreneur/:id`
3. **Authentication UI**
   - Create Login and Register pages.
   - Allow users to select role (Investor/Entrepreneur) during registration.
   - Add form validations (email format, password strength, etc.).
4. **Dashboard Layout**
   - Create a shared `DashboardLayout` component with a navbar and sidebar.
5. **Reusable UI Components**
   - Create and style reusable components: Button, InputField, Card, Avatar, etc.

Backend:

1. **Project Setup**
   - Initialize a Node.js project with Express.js.
   - Set up folder structure: `controllers`, `routes`, `models`, `middlewares`, `config`.
   - Connect to MongoDB using Mongoose.
2. **User Model**
   - Define schema: `name`, `email`, `password`, `role`, `createdAt`.
3. **Auth APIs**
   - `POST /api/auth/register`
   - `POST /api/auth/login`
   - Use bcrypt for password hashing
   - Issue JWT upon successful login
4. **Middleware**
   - Auth middleware to verify JWT tokens
   - Role-based route protection
5. **Frontend Integration**
   - Use Axios to connect login/register forms to backend APIs
   - Store JWT in localStorage and attach it in Axios headers
   - Redirect users based on role

Deliverables:

- Working login & register flow (with API integration)
- Dashboard layout implemented
- GitHub repo pushed with README

WEEK 2: Dashboards + Profiles + Collaboration Request System

Goals:

- Create investor and entrepreneur dashboards
- Implement profile views and collaboration request system

Tasks:

Frontend:

1. **Investor Dashboard Page**
    - Fetch and list entrepreneurs
    - Each entrepreneur shown in a card with: name, startup, pitch summary, and Message/Request button
2. **Entrepreneur Dashboard Page**
    - Display collaboration requests from investors
    - Show investor name, profile snippet, request status (Pending/Accepted/Rejected)
3. **Profile Pages**
    - View self and others' profiles
    - Entrepreneur profile includes: bio, startup description, funding need, pitch deck placeholder
    - Investor profile includes: bio, investment interests, portfolio companies

Backend:

1. **Profile Models**
    - Create separate schema models if necessary or extend User model
2. **Profile APIs**
    - `GET /api/profile/:id`
    - `PUT /api/profile` (update own profile)
    - `GET /api/entrepreneurs` (for investors)
    - `GET /api/investors` (for entrepreneurs)
3. **Collaboration Requests APIs**
    - Model: InvestorId, EntrepreneurId, status
    - `POST /api/request` – send request
    - `GET /api/requests` – fetch user’s requests
    - `PATCH /api/request/:id` – update request status
4. **Mock Data (if needed)**

        - Seed mock users and profiles

Deliverables:

- Both dashboards functional
- Profile views and edit capability
- Collaboration request system working end-to-end
- Responsiveness across devices

WEEK 3: Real-time Communication + Polish + Deployment

Goals:

- Implement chat feature using Socket.io
- Polish UI/UX and deploy application

Tasks:

Chat System:

1. **Backend (Socket.io):**
    - Setup Socket.io server
    - Create chat room logic (each pair gets unique room ID)
    - Emit and receive messages
    - Store messages in MongoDB:
        - Fields: senderId, receiverId, message, timestamp
2. **Chat APIs:**
    - `GET /api/chat/:userId` – get all messages between users
3. **Frontend:**
    - Create chat page: `/chat/:userId`
    - Message input field

○ Sent/received messages with timestamps and alignment ○ Show sender name/avatar
○ Optional: show online/offline mock status

UI Polish & Testing:

1. **Improvements:**
   ○ Mobile responsiveness check
   ○ Consistent colors, spacing, button styles
   ○ Add transitions, hover states
   ○ Optional: dark mode toggle
2. **Testing:**

   ○ End-to-end flow testing (auth, profile, chat)
   ○ Test all APIs via Postman
   ○ Mobile view testing

Deployment:

1. **Backend:** Deploy to Railway/Render/Heroku
2. **Frontend:** Deploy to Vercel or Netlify
3. **Environment Variables:** Setup `.env` for both frontend/backend
4. **Demo Prep:** Record 5-7 min walkthrough or prepare slides

Deliverables:

● Real-time chat system completed
● Final UI polished and responsive
● Fully deployed frontend and backend
● GitHub repository updated
● Demo video or deck ready

Optional Advanced Features:

1. **Bookmarking / Favorites:**
   ○ Allow users to save profiles
2. **Search & Filter:**
   ○ Filter profiles by industry, funding amount, location
3. **Admin Panel:**
   ○ View user stats, manage accounts, monitor chat logs
4. **Analytics:**

    - Show number of requests sent/received, chat engagement, etc.
5. **Email Invites / Notifications:**
    - Send email when request is accepted or new message received

Important Notes for Interns:

- Follow Git workflow: `feature branches` + `pull requests`
- Write clean, modular, reusable code
- Comment and document where needed
- Communicate blockers early with team leads

Deadline:

24th July 2025

Ensure all tasks are complete and code is pushed by this date.