



**Global Next Consulting India Private Limited**

**GNCIPL**

**(Leader In Consulting)**

GSTIN:09AALCG8170B1ZI

CIN: U62099UP2025PTC217716



### **Software Project Details for Client Documentation**

Section	Details
Project Title	Global_Connect – Professional Networking Platform
Client Name	Confidential
Project Manager	Ravi Kant
Team Members	Frontend Developer(s), Backend Developer(s), UI/UX Designer, QA Engineer, DevOps
Project Duration	~8–10 weeks (Timeline may vary)
Start Date	16 July 2025
End Date	05 Aug 2025
Platform Type	Web Application (MERN Stack)
Frontend Technology	React.js, Redux Toolkit, Tailwind CSS
Backend Technology	Node.js, Express.js



Section	Details
Database	MongoDB with Mongoose
Authentication	JWT, Google OAuth
Real-Time Communication	Socket.io (chat, notifications)
Storage Services	MongoDB GridFS / Cloudinary for media
Deployment	Vercel (Frontend), Render/DigitalOcean (Backend), MongoDB Atlas
Version Control	GitHub (Private/Org Repo)
Key Modules	Authentication, User Profiles, Connections, Posts, Chat, Job Board, Admin Panel
Target Audience	Job Seekers, Recruiters, Professionals
Browsers Supported	Chrome, Firefox, Safari, Edge
Devices Supported	Desktop, Tablet, Mobile (Responsive)
Project Objective	Create a scalable, secure, and user-friendly professional networking platform to connect users, facilitate job opportunities, and enable communication.
Scope of Work	Complete design, development, testing, and deployment of ProConnect MVP including admin tools
Out-of-Scope	Native mobile app, premium subscription/payment module (unless stated in scope)
Delivery Model	Agile / Iterative Development (2-week sprints)
Communication Tools	Slack, Zoom, Google Meet, Trello/Jira
Reporting Frequency	Weekly standup + Milestone Review



Section	Details
Testing Plan	Unit, Integration, End-to-End, UAT
Documentation Provided	User Manual, Admin Manual, Deployment Guide, API Docs, Wireframes, Release Notes
Post Deployment Support	30 Days (as per SLA)
Client Responsibilities	Content inputs, domain/DNS access, UAT participation
Success Criteria	UAT sign-off, functional stability, performance within defined benchmarks

#### Optional: Attachments (for appendices or annexures)

Attachment No.	Document
Annexure A	UI Wireframes (PDF or Figma)
Annexure B	Database ER Diagram
Annexure C	API Endpoint List
Annexure D	Project Timeline (Gantt Chart)
Annexure E	Change Request Template
Annexure F	Risk Register

#### **Project Name: Global\_Connect – Professional Networking Platform**

### **1. Project Overview**

A full-stack professional networking platform enabling users to build a profile, connect with others, share updates, chat in real time, and apply or post jobs —



## 2. Goals & Objectives

Goal	Details
Networking	Users connect with professionals in their domain.
Career Advancement	Browse, apply to, or post jobs.
Engagement	Users post updates, react, and comment.
Communication	Real-time chat between connected users.
Admin Supervision	Admin manages reports, users, and job postings.

---

## 3. Architecture Overview

- **Frontend:** React + Redux Toolkit + Tailwind CSS
- **Backend:** Node.js + Express.js
- **Database:** MongoDB + Mongoose
- **Authentication:** JWT + Google OAuth
- **Real-Time Chat:** Socket.IO
- **File Uploads:** Multer (profile images, post media)
- **Deployment:** Vercel (frontend) + Render/DigitalOcean (backend)
- **CI/CD (Optional):** GitHub Actions or Docker

---

## 4. Major Modules & Features

### A. Authentication

- Sign up, login (JWT + bcrypt)
- Google OAuth login
- Password reset via email
- Middleware for auth guard

### B. User Profiles

- View/edit profile
- Upload profile picture and banner
- Add experience, skills, education

### C. Connections



- Send connection requests
- Accept/reject
- List of connections

#### **D. Feed/Posts**

- Create text/image post
- Like, comment, repost
- View own + connection feed

#### **E. Chat/Messaging**

- Real-time chat via Socket.io
- Unread indicators
- Chat history

#### **F. Job Board**

- Post job (title, desc, skills, location)
- Search/apply for jobs
- Save job, view application status

#### **G. Search & Filters**

- Global search (user, jobs, posts)
- Filters (skills, location, experience)

#### **H. Notifications**

- New connection, message, job update
- Real-time and persistent notifications

#### **I. Admin Panel**

- Manage users, jobs, posts
- Handle reported content

---

## 5. Database Schema Design (ERD Summary)

### **User**

```
{
  _id, name, email, password, bio, profilePic,
  experience: [{company, role, from, to}],
  education: [{school, degree, from, to}],
}
```



```

skills: [String],
connections: [ObjectId],
connectionRequests: [ObjectId],
savedJobs: [ObjectId]

```

```

}

```

### Post

```

{
  _id, userId, content, image, createdAt,
  likes: [ObjectId],
  comments: [{userId, text, createdAt}]
}

```

### Message

```

{
  _id, senderId, receiverId,
  content, timestamp
}

```

### Job

```

{
  _id, postedBy, title, description,
  skills: [String], applicants: [ObjectId],
  createdAt
}

```

## 6. API Endpoints (Backend - Express.js)

Method	Endpoint	Description
POST	/api/auth/register	Register user
POST	/api/auth/login	Login user
GET	/api/users/:id	Get user profile
PUT	/api/users/:id	Update profile
POST	/api/posts/	Create post



Method	Endpoint	Description
GET	/api/posts/feed/:userId	Get user feed
POST	/api/messages/	Send message
GET	/api/messages/:sender/:receiver	Get chat history
POST	/api/jobs/	Post a job
GET	/api/jobs/	List all jobs
POST	/api/jobs/apply/:jobId	Apply to job

## 7. Testing Plan

- **Unit Tests** (Jest): Functions, services, APIs
- **Integration Tests** (Supertest): Route testing
- **Frontend Testing** (React Testing Library)
- **E2E Testing (Optional)**: Cypress

## 8. Dev Tools & Libraries

Tool	Purpose
Multer	File uploads
JWT	Authentication
Bcrypt	Password hashing
Mongoose	MongoDB ORM
Socket.io	Real-time chat
Redux Toolkit	State management
Tailwind CSS	Styling
Dotenv	Environment config



## 9. Deployment Strategy

### Frontend:

- Deploy via **Vercel** or **Netlify**
- Environment: REACT\_APP\_API\_URL

### Backend:

- Use **Render**, **Railway**, or **DigitalOcean**
  - Set env vars: JWT\_SECRET, DB\_URI, etc.
  - Use PM2 or Docker (optional)
- 

## 10. Documentation & Guides

### Developer Guide

- Environment setup
- API usage & auth headers
- How to run frontend/backend

### Admin Guide

- Reset user passwords
- Manage reports
- Job moderation

### User Guide

- How to complete profile
  - Make connections
  - Apply for jobs
- 

## 11. Team Roles (for group project)

Role	Responsibility
Frontend Lead	React components, UI/UX
Backend Lead	APIs, DB schema, logic
Auth/Chat Dev	JWT/OAuth + messaging
QA/Tester	Manual + automated tests



Role	Responsibility
DevOps	Deployment & CI/CD
Documentation Lead	README, guides, diagrams

## 12. Suggested Timeline

Week	Milestone
Week 1	Project setup, auth, DB models
Week 2	User profiles, connection logic
Week 3	Posts, likes, comments
Week 4	Chat system + notifications
Week 5	Job board + search
Week 6	Admin panel + polish + deploy

## Final Deliverables

- Working app (deployed)
- GitHub repo with clear README
- API documentation (Postman collection)
- Architecture diagrams (ERD, flowcharts)
- Deployment scripts (Dockerfile / setup.md)
- Optional: Video demo + PDF report

## DETAILED CLIENT DOCUMENT LIST FOR FULL-STACK LINKEDIN-LIKE PLATFORM

### PHASE 1: INITIATION & DISCOVERY

#	Document Name	Description
1.1	Project Proposal	Overview of solution, features, benefits, business goals, and rough cost



#	Document Name	Description
1.2	<b>Client Requirements Brief</b>	Client-stated needs, goals, workflows, pain points
1.3	<b>Feasibility Study</b>	Technical, operational, and financial feasibility assessment
1.4	<b>Project Charter</b>	Authorizes the project, identifies PM, goals, budget, timeline, authority levels
1.5	<b>Stakeholder Register</b>	List of stakeholders, roles, contact, influence/impact matrix
1.6	<b>Risk Assessment (Initial)</b>	Risks, mitigation, and responsibilities before development starts

## PHASE 2: PLANNING & ANALYSIS

#	Document Name	Description
2.1	<b>Business Requirements Document (BRD)</b>	What the business wants from the platform (in layman terms)
2.2	<b>System Requirements Specification (SRS)</b>	Full detailed list of functional and non-functional requirements
2.3	<b>Scope of Work (SOW)</b>	Exact scope of delivery, out-of-scope, timeline, deliverables
2.4	<b>Work Breakdown Structure (WBS)</b>	Hierarchical decomposition of project tasks
2.5	<b>Project Plan</b>	Timeline, resources, milestones, Gantt chart
2.6	<b>Communication Plan</b>	Who communicates what to whom and when (client/PM/devs)
2.7	<b>Change Management Plan</b>	Process for accepting/modifying features during development
2.8	<b>Acceptance Criteria Document</b>	Client-agreed criteria for accepting each



#	Document Name	Description
		major feature/module
2.9	Legal Documents	NDA, IP agreement, licensing, support & maintenance contract

### PHASE 3: DESIGN

#	Document Name	Description
3.1	Wireframes / Mockups	Lo-fi or hi-fi wireframes of key screens
3.2	UI/UX Design Specification	Color palette, typography, component styles, interaction
3.3	Information Architecture (IA)	Navigation structure, user flow diagrams
3.4	ER Diagram / DB Design	Database schema, tables/collections, relationships
3.5	System Architecture Diagram	Component and data flow diagram (frontend-backend)
3.6	Data Privacy Policy	How user data will be handled, stored, processed securely

### PHASE 4: DEVELOPMENT

#	Document Name	Description
4.1	API Specification (Swagger/Postman)	List of API endpoints, methods, parameters, responses
4.2	Backend Functional Map	Mapping of backend features/modules and responsible services
4.3	Authentication Flow Diagram	JWT/OAuth-based login/registration/session flow
4.4	Feature/Module Development	Client-facing tracker of completed modules



#	Document Name	Description
	Checklist	
4.5	Third-party Integration Notes	Notes on any APIs/tools used: Google OAuth, Socket.io, Stripe
4.6	Code Handoff Guide (if client takes over)	Developer setup guide, folder structure, dependencies

## PHASE 5: TESTING & QA

#	Document Name	Description
5.1	Test Plan	Overall test strategy, tools, coverage
5.2	Test Case Document	All test scenarios with expected outcomes
5.3	Bug Tracker Log	Issues found during UAT, internal QA (shared via Excel/Jira)
5.4	Security Test Report	Pen testing results, secure headers, CSRF/XSS protections
5.5	Performance Test Report	Page speed, DB queries, load times, optimization notes
5.6	User Acceptance Testing (UAT) Checklist	Final checklist for client approval with sign-off
5.7	Accessibility Compliance Report	WCAG 2.1 AA-level features, tested assistive tools

## PHASE 6: DEPLOYMENT & DELIVERY

#	Document Name	Description
6.1	Deployment Guide	Step-by-step deployment instructions for frontend/backend
6.2	Environment Configuration	.env variables, Firebase config, S3 buckets, API



#	Document Name	Description
	<b>Docs</b>	keys
6.3	<b>Release Notes</b>	What's included in each version (v1.0, v1.1...)
6.4	<b>Rollback Plan</b>	What to do if something fails post-deployment
6.5	<b>Production Access Credentials Sheet</b>	Hostnames, login, SSH, DB admin creds (securely shared)
6.6	<b>Training Manual or Video</b>	Admin or user training materials
6.7	<b>Post-Deployment Checklist</b>	DNS, cache, performance, broken links, final QA
6.8	<b>Support and Maintenance Plan</b>	SLA, support hours, patching window, contacts
6.9	<b>Project Closure Report</b>	Summary of delivery, status, lessons learned
6.10	<b>Handover Certificate</b>	Signed confirmation that all scope is delivered and accepted



## OPTIONAL DELIVERABLES

#	Document Name	Description
O1	SEO Optimization Plan	Meta tags, sitemap, robots.txt, structured data
O2	Analytics & Tracking Plan	Google Analytics, Hotjar, tracking KPIs
O3	GDPR/Data Retention Policy	Right to forget, data storage & logging practices
O4	Backup & Recovery Plan	DB backup, snapshot strategy, restore steps
O5	Client Feedback Form	Post-project feedback and testimonial form
O6	SLA Monitoring Dashboard (if	Monthly/weekly issue tracking and



#	Document Name	Description
	support included)	resolution times

## Suggested Drive/Repo Folder Organization

/Client-Docs

|

└─ 1\_Initiation/

└─ 2\_Planning/

└─ 3\_Design/

└─ 4\_Development/

└─ 5\_Testing/

└─ 6\_Deployment/

└─ 7\_Support/

└─ Assets/





