








# SKILLINK - COMPLETE PROJECT ROADMAP

## Project Structure

### 📁 SkillLink (Main Project Directory)

```
|
|
| — 📁 client (Frontend - React/Next.js)
|   |
|   | — 📁 public
|   |   |
|   |   | — index.html      # Root HTML file for React App
|   |   | — favicon.ico     # Website Icon
|   |   | — manifest.json   # Metadata for PWA support
|   |   | — robots.txt     # Search Engine Instructions
|   |   | — logo.png        # Logo for website branding
|   |   |
|   |   | — 📁 src (Main Source Code)
|   |   |   |
|   |   |   | — 📁 assets      # Static assets (images, fonts, icons, etc.)
|   |   |   | — 📁 components  # Reusable UI components
|   |   |   |   |
|   |   |   |   | — Navbar.jsx  # Navigation Bar Component
|   |   |   |   | — Sidebar.jsx  # Sidebar Component
|   |   |   |   | — Footer.jsx  # Footer Component
|   |   |   |   | — Button.jsx  # Reusable Button Component
|   |   |   |   | — Loader.jsx  # Loading Animation Component
|   |   |   |   |
|   |   |   |   | — 📁 pages      # Full-page components
|   |   |   |   |   |
|   |   |   |   |   | — Home.jsx   # Home Page
|   |   |   |   |   | — Login.jsx  # Login Page
|   |   |   |   |   | — Register.jsx # Registration Page
|   |   |   |   |   | — Dashboard.jsx # Freelancer Dashboard
|   |   |   |   |   | — Profile.jsx  # User Profile Page
|   |   |   |   |   | — ProjectList.jsx # List of Projects
|   |   |   |   |   | — Chat.jsx    # Chat System Page
|   |   |   |   |   | — Payment.jsx  # Payment Page
|   |   |   |   |   | — AdminPanel.jsx # Admin Panel for User Approval
|   |   |   |   |   | — Settings.jsx  # Settings & Account Management
|   |   |   |   |   | — NotFound.jsx  # 404 Page Not Found
|   |   |   |   |   |
|   |   |   |   | — 📁 context      # React Context API logic for state management
|   |   |   |   |   |
|   |   |   |   |   | — AuthContext.js  # Manages authentication state (login/logout)
|   |   |   |   |   | — ThemeContext.js  # Manages light/dark theme
|   |   |   |   |   |
|   |   |   |   |   |
```

# SKILLINK - COMPLETE PROJECT ROADMAP

			—  hooks	# Custom React hooks (reusable functions)
				— useAuth.js # Hook for authentication logic
				— useFetch.js # Hook for API fetching
				— usePayment.js # Hook for handling payments
			—  styles	# Global and component-specific CSS files
				— index.css # Global styles
				— theme.css # Dark/Light mode styles
				— Dashboard.css # Dashboard page styles
			—  utils	# Helper functions
				— api.js # Handles API requests
				— formatDate.js # Date formatting utility
				— authHelper.js # Authentication helper functions
			— App.jsx	# Main React Component (Routes are defined here)
			— index.js	# Renders React App into DOM
			— routes.js	# Defines frontend routes
			— .env	# Environment variables (API keys, etc.)
			— package.json	# Dependencies & scripts for React
			— next.config.js	# Next.js configuration file (if using Next.js)
			— vite.config.js	# Vite configuration file (if using Vite)
			— README.md	# Documentation for frontend
			—  server (Backend - Node.js & Express)	
				—  config # Configuration files
				— dbConfig.js # MongoDB database connection
				— keys.js # API keys & secrets
			—  models	# Database schemas
				— User.js # User schema (name, email, password, role)
				— Project.js # Project schema (title, description, budget)
				— Chat.js # Chat schema (messages, timestamps)
				— Payment.js # Payment schema (transactions, status)
			—  routes	# API endpoints

# SKILLINK - COMPLETE PROJECT ROADMAP

			authRoutes.js	# Authentication routes (login, register)
			projectRoutes.js	# Project-related API endpoints
			chatRoutes.js	# Chat system API
			paymentRoutes.js	# Payment processing API
			📁 controllers	# Logic for API requests
			authController.js	# Handles user login/register
			projectController.js	# Handles project-related operations
			chatController.js	# Manages chat functionality
			paymentController.js	# Handles payments & transactions
			📁 middleware	# Security middleware
			authMiddleware.js	# Checks if user is authenticated (JWT)
			errorHandler.js	# Handles API errors
			📁 uploads	# Stores uploaded files (profile pictures, project files)
			server.js	# Main backend file (starts Express server)
			package.json	# Dependencies & scripts for Node.js
			.env	# Backend environment variables
			README.md	# Documentation for backend
			📁 docs (Documentation)	
			API_Documentation.md	# API usage guide
			Database_Schema.md	# Database structure details
			Project_Guide.md	# Project setup & deployment instructions
			.gitignore	# Ignores unnecessary files for Git
			README.md	# Overall project documentation

## Key Features & Tech Stack

- **Frontend:** Next.js (React.js for UI/UX)
- **Backend:** Node.js (Express.js for REST API)
- **Database:** MongoDB (Mongoose ORM)
- **Authentication:** Firebase Auth (Google, Facebook, LinkedIn, Email)
- **Payments:** PayPal, Razorpay, Paytm API (Supporting USD & INR)

# SKILLINK - COMPLETE PROJECT ROADMAP

- **Hosting:** Vercel (Frontend) + AWS/DigitalOcean (Backend)
- **AI Chatbot:** OpenAI GPT API for Smart Help Desk

## Implementation Plan

1. **Setup the Environment:** Install Node.js, MongoDB, Git, VS Code, and required dependencies.
2. **Develop Frontend:** Implement Next.js UI pages & components with routing.
3. **Develop Backend:** Create REST API, models, authentication, and middleware.
4. **Integrate Payment Gateway:** Implement Razorpay & PayPal.
5. **Deploy & Test:** Deploy Frontend on Vercel & Backend on AWS/DigitalOcean.
6. **Optimize & Debug:** Ensure full functionality without errors.