



**INSTITUTE FOR ADVANCED COMPUTING AND
SOFTWARE DEVELOPMENT, AKURDI, PUNE**

“Connexus”

PG-DACAUGUST 2025

Submitted By:

Group No: 48

Roll No.	Name of Student
258058	Pranav Patil
258176	Raj Dhakar

Mrs. Neelam
Project Guide

Mr. Anil Sharma
Centre Coordinator

ABSTRACT

Connexus is a high-performance, scalable professional networking platform designed to facilitate seamless interaction, content sharing, and real-time communication. Leveraging a modern Microservices Architecture, the platform integrates complex networking logic with real-time social features to provide a robust user experience tailored for professional networking.

The platform's frontend is built using React and Material UI, ensuring a responsive, premium, and intuitive user interface with smooth animations powered by Framer Motion. It utilizes StompJS and WebSockets to provide real-time messaging and instant notifications.

On the backend, Connexus employs a suite of Spring Boot Microservices, including dedicated services for User Management, Posts, Connections, Messaging, and Notifications. Key architectural components include Netflix Eureka for service discovery and a centralized API Gateway for secure routing. Security is enhanced through Spring Security with JWT-based authentication, while media assets are managed and delivered via Cloudinary. By decoupling core functionalities into independent services, Connexus achieves high scalability and fault tolerance, making it a robust for professional communities.

ACKNOWLEDGEMENT

I take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination. I extend my heartfelt thanks to our esteemed guide, **Mrs.Neelam** for providing me with the right guidance and advice at the crucial juncture and showing me the right way. I sincerely thank our respected Centre Co-Ordinator, Mr.Anil sharma, for allowing us to use the available facilities. I would also like to thank the other faculty members at this occasion. Last but not least, I would like to thank my friends and family for the support and encouragement they have given me during our work.

Pranav Patil (240341220055)

Raj Dhakar (2403412200156)

Table of Contents

Sr.No	Description	Page No.
1	Introduction	1
2	SRS	9
3	Diagrams	15
3.1	ER Diagram	15
3.2	Use Case Diagram	16
3.3	Data Flow Diagram	17
3.4	Activity Diagram	18
3.5	Class Diagram	21
3.6	Component Diagram	22
3.7	Sequence Diagram	23
4	Database Design	24
5	Snapshots	27
6	Conclusion	32
7	References	33

1. INTRODUCTION

Connexus is a high-performance professional networking and social interaction platform designed to bridge the gap between career opportunities and professional community engagement. It is a comprehensive system built on a Microservices Architecture, integrating advanced networking logic with real-time social functionalities. Connexus allows professionals to showcase their expertise, share industry insights through multimedia posts, and build meaningful professional relationships through a secure and scalable environment.

The platform utilizes React and Material UI for a sophisticated, responsive frontend that prioritizes user experience with smooth animations via Framer Motion. The backend is powered by a robust suite of Spring Boot Microservices, ensuring high availability and fault tolerance. Security is a core pillar of the platform, implemented through Spring Security for fine-grained authentication and authorization, utilizing JWT for stateless session management. Real-time features, such as instant messaging and notifications, are facilitated by WebSockets (StompJS).

Connexus is tailored to meet the needs of the modern workforce, providing tools for both job seekers (Professionals) and talent scouts (Recruiters), while maintaining a secure, role-governed ecosystem managed by system Administrators

1.1 Roles and Access Control

In the Connexus system, distinct roles—Admin, Professional, and Recruiter—ensure that tasks are appropriately delegated and the platform remains secure.

1. Admin Role

The Admin role has comprehensive control over the platform's infrastructure and community standards. Admins are responsible for the health and integrity of the professional network.

❖ Responsibilities:

- **User & Content Auditing:** Monitor platform activity, manage user reports, and moderate content to ensure professional standards.
- **Service Monitoring:** Oversee the status of the various microservices (via Eureka) and the API Gateway.
- **System Configuration:** Manage global settings, including security policies and third-party integrations (Cloudinary, etc.).
- **Reporting & Analytics:** Generate insights on user engagement, network growth, and platform performance.
- **Access Level:**

Full access to administrative dashboards and system logs.

Ability to manage user permissions and handle security-related overrides.

2. Professional (User) Role

The Professional role is the core of the community, focusing on networking, content creation, and career growth.

❖ **Responsibilities:**

- **Profile Management:** Create and update a professional digital identity, showcasing skills, experience, and education.
- **Networking:** Send and receive connection requests, build a professional network, and interact with peers.
- **Content Sharing:** Create and engage with posts (text and media), follow industry trends, and participate in professional discussions.
- **Real-time Communication:** Securely message connections and receive live notifications for network activity.
- **Access Level:**

Access to personalized feeds, messaging modules, and network management tools.

No access to system configurations or other users' private administrative data.

Access Level:

Full access to administrative dashboards and system logs.

Ability to manage user permissions and handle security-related overrides.

2. Professional (User) Role

The Professional role is the core of the community, focusing on networking, content creation, and career growth.

Responsibilities:

Profile Management: Create and update a professional digital identity, showcasing skills, experience, and education.

Networking: Send and receive connection requests, build a professional network, and interact with peers.

- **Content Sharing:** Create and engage with posts (text and media), follow industry trends, and participate in professional discussions.
- **Real-time Communication:** Securely message connections and receive live notifications for network activity.
- **Access Level:**

Access to personalized feeds, messaging modules, and network management tools.

No access to system configurations or other users' private administrative data.

❖ **Recruiter Role**

The Recruiter role is a specialized user type focused on talent acquisition and candidate engagement.

- **Responsibilities:**

Talent Discovery: Access specialized dashboards to find and filter professional talent.

Engagement: Communicate with potential candidates and manage recruitment-related interactions.

Dashboard Management: Utilize the Recruiter Dashboard to track engagement and talent pipelines.

- **Access Level:**

Access to recruitment-specific tools and dashboards.

Permissions to view professional profiles and initiate professional outreach.

2. Professional (User) Role

The Professional role is the core of the community, focusing on networking, content creation, and career growth.

Responsibilities:

Profile Management: Create and update a professional digital identity, showcasing skills, experience, and education.

Networking: Send and receive connection requests, build a professional network, and interact with peers.

Content Sharing: Create and engage with posts (text and media), follow industry trends, and participate in professional discussions.

Real-time Communication: Securely message connections and receive live notifications for network activity.

Access Level:

Access to personalized feeds, messaging modules, and network management tools.

No access to system configurations or other users' private administrative data.

3. Recruiter Role

The Recruiter role is a specialized user type focused on talent acquisition and candidate engagement.

Responsibilities:

Talent Discovery: Access specialized dashboards to find and filter professional talent.

Engagement: Communicate with potential candidates and manage recruitment-related interactions.

Dashboard Management: Utilize the Recruiter Dashboard to track engagement and talent pipelines.

Access Level:

Access to recruitment-specific tools and dashboards.

Permissions to view professional profiles and initiate professional outreach.

1.2 Purpose

The purpose of Connexus is to provide a unified, modern environment where professionals can connect, communicate, and grow. By leveraging a microservices architecture, the system aims to solve the scalability issues of traditional monolithic social platforms while providing a highly responsive and secure user experience. It streamlines professional networking by integrating real-time messaging, media sharing, and talent acquisition into a single cohesive platform.

1.3 Scope

The scope of the Connexus project encompasses the full-stack development of a microservices-based social ecosystem. This includes:

Developing independent services for Users, Posts, Connections, Messaging, and Notifications.

Implementing a centralized API Gateway and Service Discovery (Eureka).

Creating a premium React-based frontend with a focus on professional aesthetics.

Integrating Cloudinary for scalable media storage and WebSockets for real-time delivery.

1.4 Objectives of Connexus

Distributed Scalability: Utilize microservices to allow independent scaling of high-traffic features like messaging and posts.

Real-Time Engagement: Provide sub-second latency for chats and notifications via persistent WebSocket connections.

Robust Security: Ensure data privacy and secure communication using JWT and Spring Security.

Professional UI/UX: Deliver a high-end interface that feels premium and promotes professional interaction.

Seamless Media Integration: Support multimedia post creation with efficient cloud storage.

Reliable Networking Logic: Implement complex connection algorithms to manage professional relationships effectively.

1.5 Functionalities Provided by Connexus

1. Identity & User Management

Secure Authentication: JWT-based login and registration with encrypted password storage.

Granular Profiles: Detailed professional profiles with media support via Cloudinary.

Role-Based Access (RBAC): Strict enforcement of Admin, Professional, and Recruiter permissions.

2. Content & Post Management

Multimedia Sharing: Create posts with text and images, managed via the post-service.

Interactive Feeds: Real-time social feed with cascading updates and interactive engagement.

Activity Tracking: Monitor "My posts" and network interactions through a dedicated UI.

❖ Professional Networking (Connection Service)

- **Invitation System:** Manage incoming and outgoing professional invitations.
- **Network Mapping:** Track professional relationships and suggestions.
- **Dynamic Actions:** Accept, ignore, or remove connections with real-time state updates.

❖ 4. Real-Time Messaging & Notifications

- **Instant Chat:** Peer-to-peer messaging using the messaging-service and WebSockets.
- **Live Notifications:** Real-time alerts for messages, connections, and post interactions.
- **Message Persistence:** Storing and retrieving chat history across sessions.

❖ 5. Specialized Recruiter Tools

- **Recruiter Dashboard:** Centralized hub for managing talent discovery and recruitment workflows.
- **Candidate Search:** Enhanced visibility into professional talent within the network.

❖ 6. Architectural Resilience

- **Service Discovery:** Automated registration and discovery of services via Eureka.
- **Gateway Routing:** Single entry point for all frontend requests, simplifying client-side API logic.
- **Fault Tolerance:** Decoupled services ensure that a failure in one module (e.g., Notifications) doesn't crash the entire platform.

2. SOFTWARE REQUIREMENT SPECIFICATION

The functional requirements for Connexus define the specific features and technical capabilities that the system must provide to support a robust professional networking ecosystem. These requirements ensure that the microservices architecture effectively addresses user needs and business objectives.

2.1 Functional Requirements for Connexus

1. Identity and User Management

- **User Registration and Login:**

The system shall allow users to create professional accounts and authenticate securely using JWT-based login mechanisms

- **Role-Based Access Control (RBAC):**

The system shall enforce distinct permissions for Professionals, Recruiters, and Admins to ensure data privacy and operational integrity.

- **Profile Management:**

Users shall be able to build and update detailed professional profiles, including designations, skills, professional summaries, and profile images hosted via Cloudinary.

2. Post and Content Management

- **Multi-Media Post Creation:**

The system shall allow users to publish posts containing titles, long-form descriptions, and media attachments (images/videos).

- **Interactive Professional Feed:**

The system shall provide a dynamic feed of posts from the user's network, supporting real-time interaction (likes, comments)

- **Personal Activity Tracking:**

Users shall have a dedicated "My Posts" section to view and manage their historical content..

3. Professional Networking (Connection Service)

- **Connection Requests:**

Users shall be able to send, receive, and manage professional connection requests to expand their network

- **Invitation Management:**

The system shall provide a centralized interface for users to "Accept" or "Ignore" incoming invitations.

- **Network Visibility:**

The system shall allow users to view their current professional network and discover potential connections.

4. Real-Time Messaging and Notifications

- **Instant Peer-to-Peer Chat:**

The system shall support real-time messaging between connections using WebSockets (StompJS).

- **Live Notifications:**

The system shall deliver instant alerts for new connection requests, messages, and social interactions.

- **Message History:**

The system shall persist chat history, allowing users to retrieve previous conversations across sessions.

5. Specialized Recruiter Dashboard

- **Talent Search & Discovery:**

Recruiters shall have access to advanced search tools to find professional talent based on skills, location, and experience.

- **Recruitment Management:**

Recruiters shall have access to advanced search tools to find professional talent based on skills, location, and experience.

6. Security & Infrastructure

- **Stateless Authentication:**

The system shall use JWT for secure, stateless communication between the frontend and microservices.

- **Service Discovery & Routing:**

The system shall utilize Netflix Eureka for service discovery and an API Gateway for centralized request management and security.

2.2 Non-Functional Requirements for Inventory SellMart

1. Performance

- **Response Time:**
The system shall respond to UI Interaction within 2 seconds under normal operating conditions.
- **Scalability:**
The microservices architecture shall support horizontal scaling to handle increasing user loads and data volume.
- **Real-time Latency:**
Messaging and notification delivery shall maintain sub-second latency via persistent WebSocket connections.

2. Reliability & Availability

- **Premium UI Design:**
The interface shall be built with Material UI to provide a clean, professional, and intuitive user experience.
- **Responsiveness:**
The platform shall be fully responsive, ensuring seamless operation across desktops, tablets, and smartphones.
- **Error Handling:**
The system shall gracefully handle errors and provide meaningful feedback to users when issues occur.

3. Usability

- **User Interface:**
The system shall have a user-friendly interface that is easy to navigate, with clear instructions and minimal learning curve.

4. Maintainability

- **Modularity:**
The system shall be designed in a modular fashion, allowing for easy updates and enhancements to individual components without affecting the entire system.

- **Other Requirements:**

Hardware and Network Interfaces:**Back-end Server Configuration:**

- Processor: Multi-core (Intel i5/i7 or equivalent)
- RAM: 16 GB (Recommended for running multiple microservices)

Client Configuration:

- Processor: Modern dual-core/quad-core
- RAM: 8 GB+
- Display: 1080p resolution for optimal UI experience

Software Interfaces:

Software configuration for back-end Services:

- Java 17+
- Spring Boot (Auth, JPA, Gateway, Eureka)
- MySQL (User & Post data)
- Neo4j (Graph-based Connection data)

Software configuration for front-end Services:

- ReactJs
- Material UI
- Axios
- StompJs (WebSocket Client)

External Services:

- Cloudinary (media Storage API)
- JWT (Security Token Standard)
- StompJs (WebSocket Client)

3. DIAGRAMS

3.1 Entity Relationship Diagram:

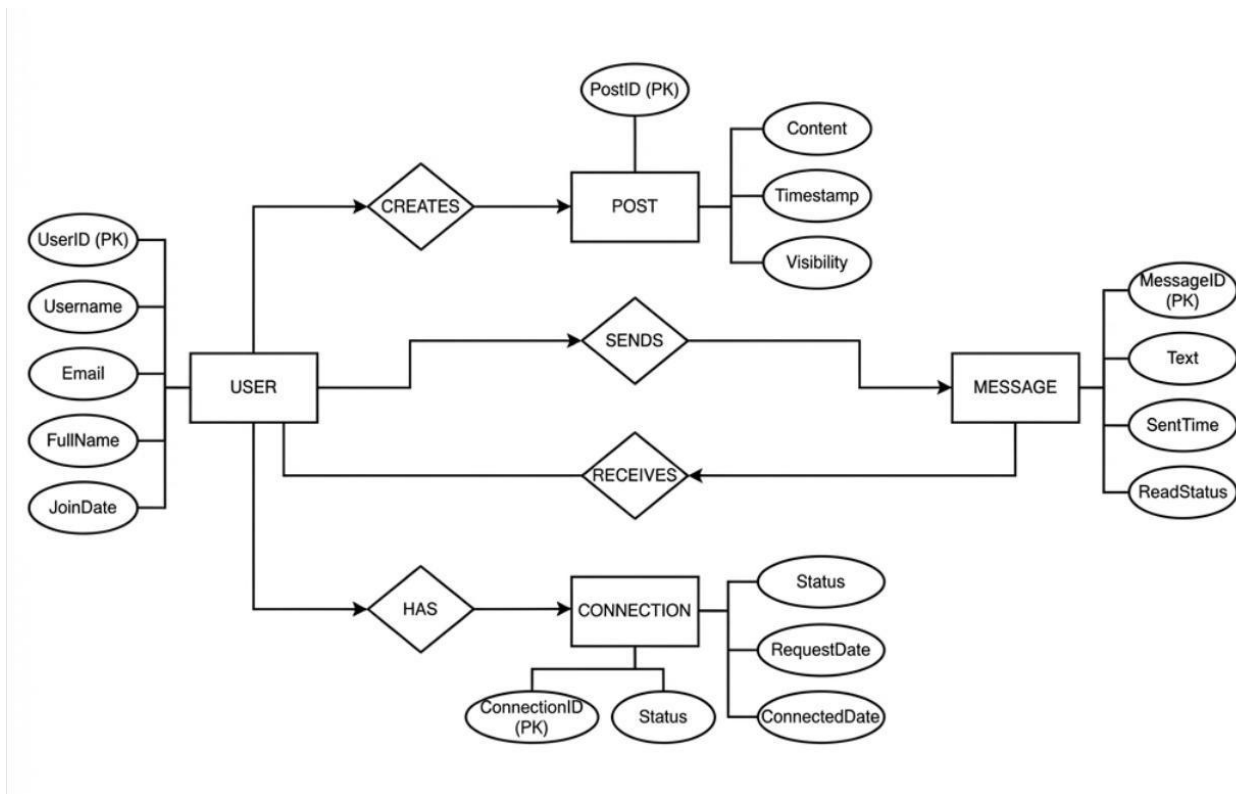


Fig. ER Diagram for Connexus

3.2 Use Case Diagram:

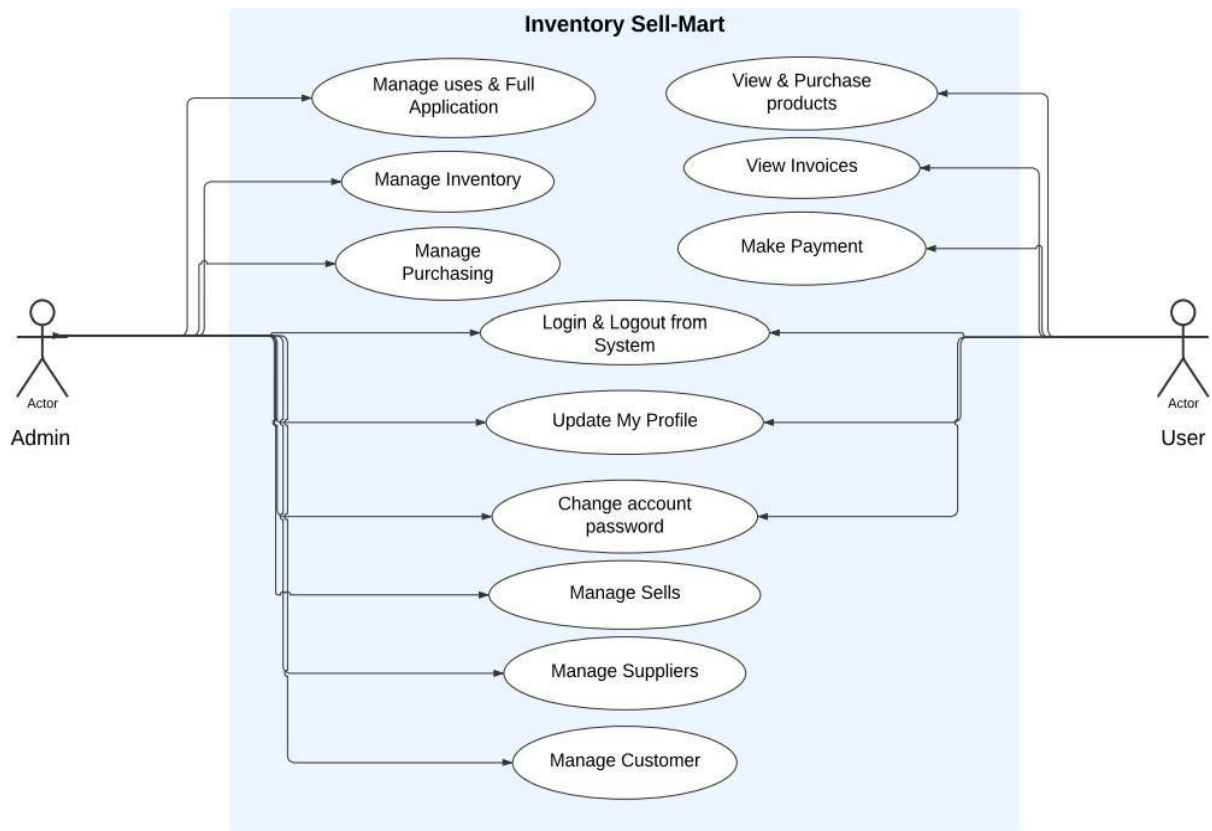
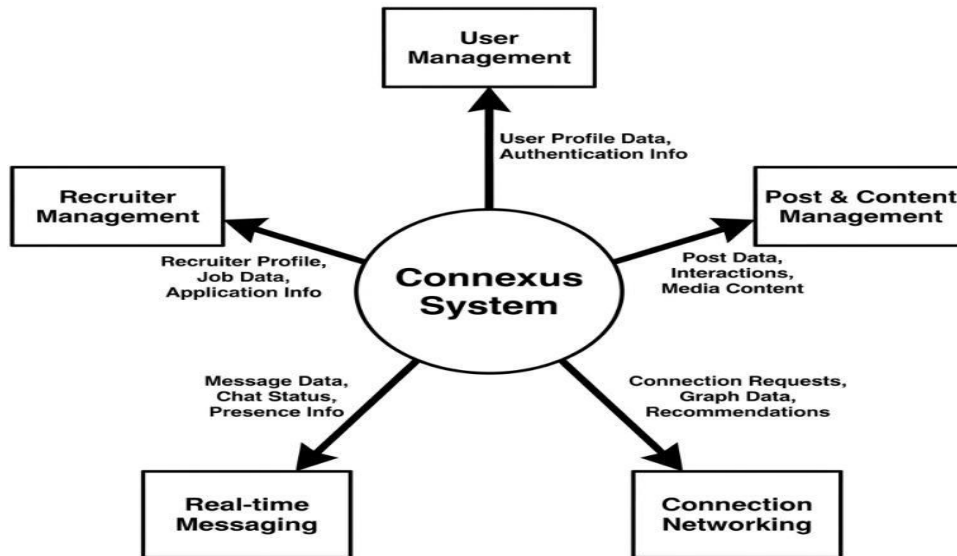


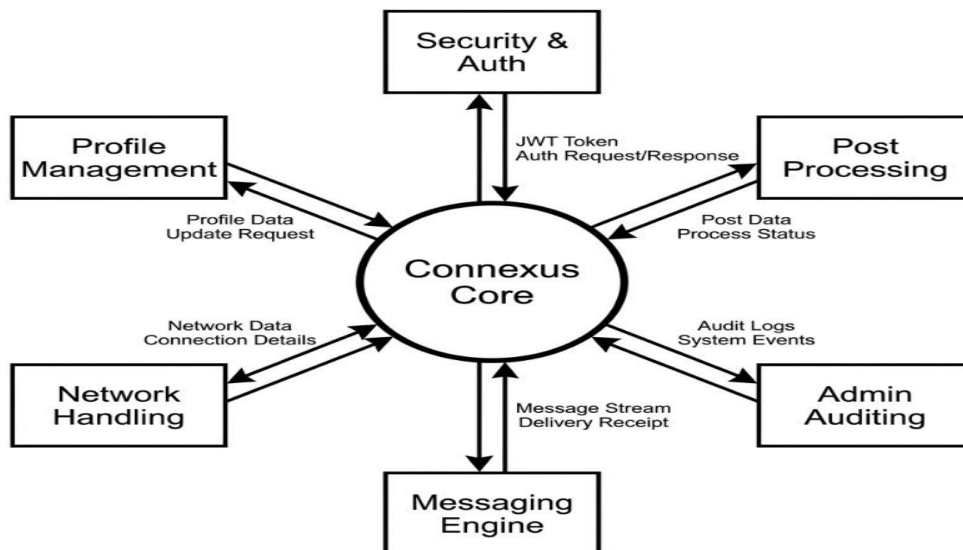
Fig. Use Case Diagram for Connexus

3.3 Data Flow Diagram:

DFD Level 0:



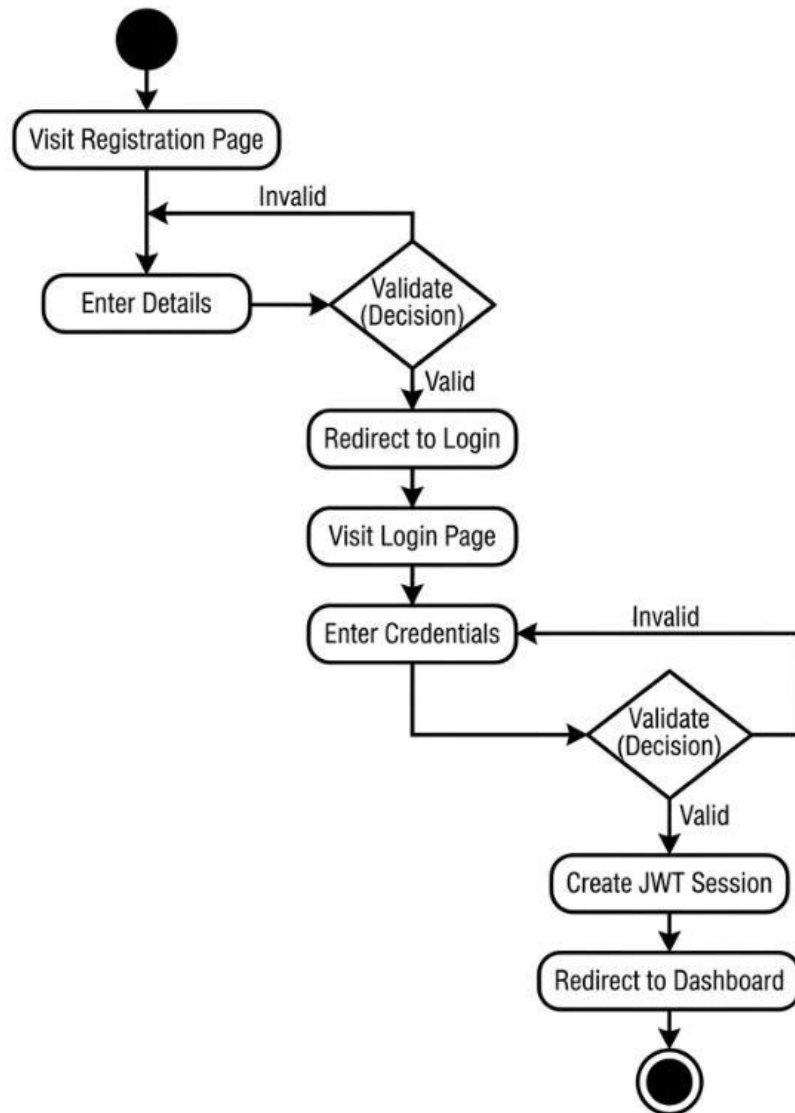
DFD level 1



3.4 Activity Diagram :

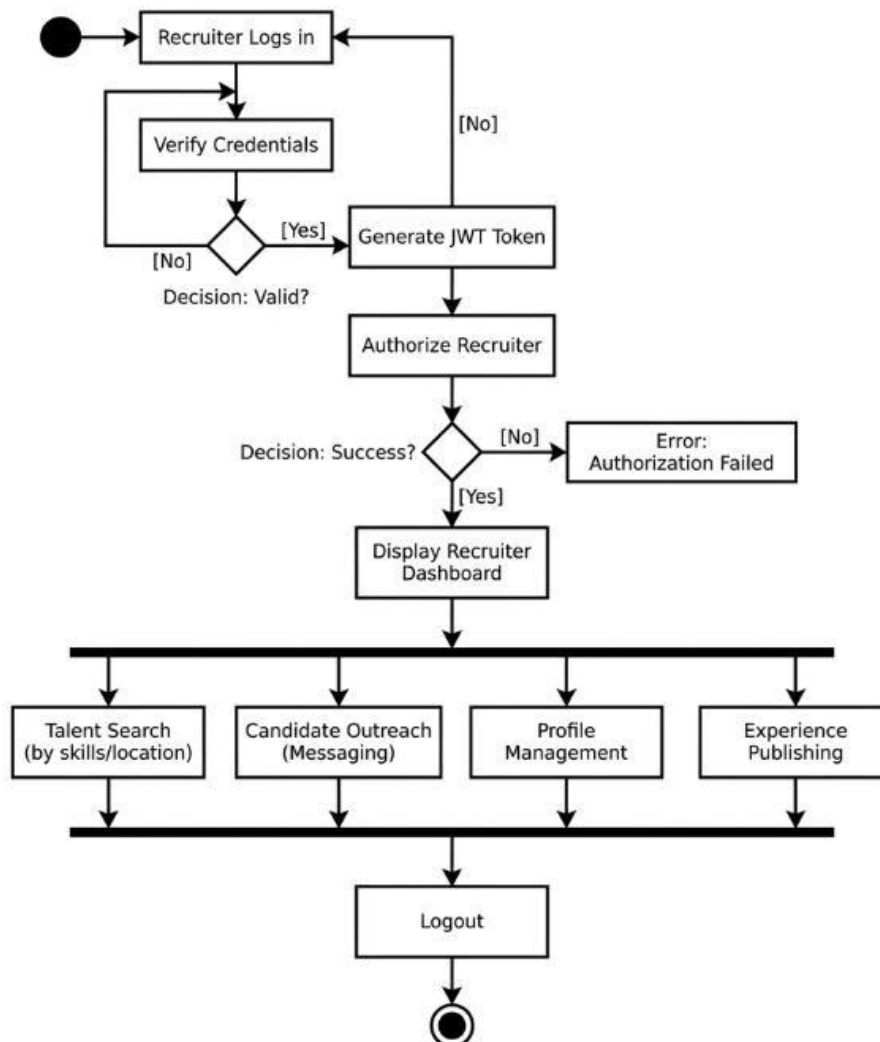
1. Login Activity Diagram

Connexus: Login and Registration Activity Diagram



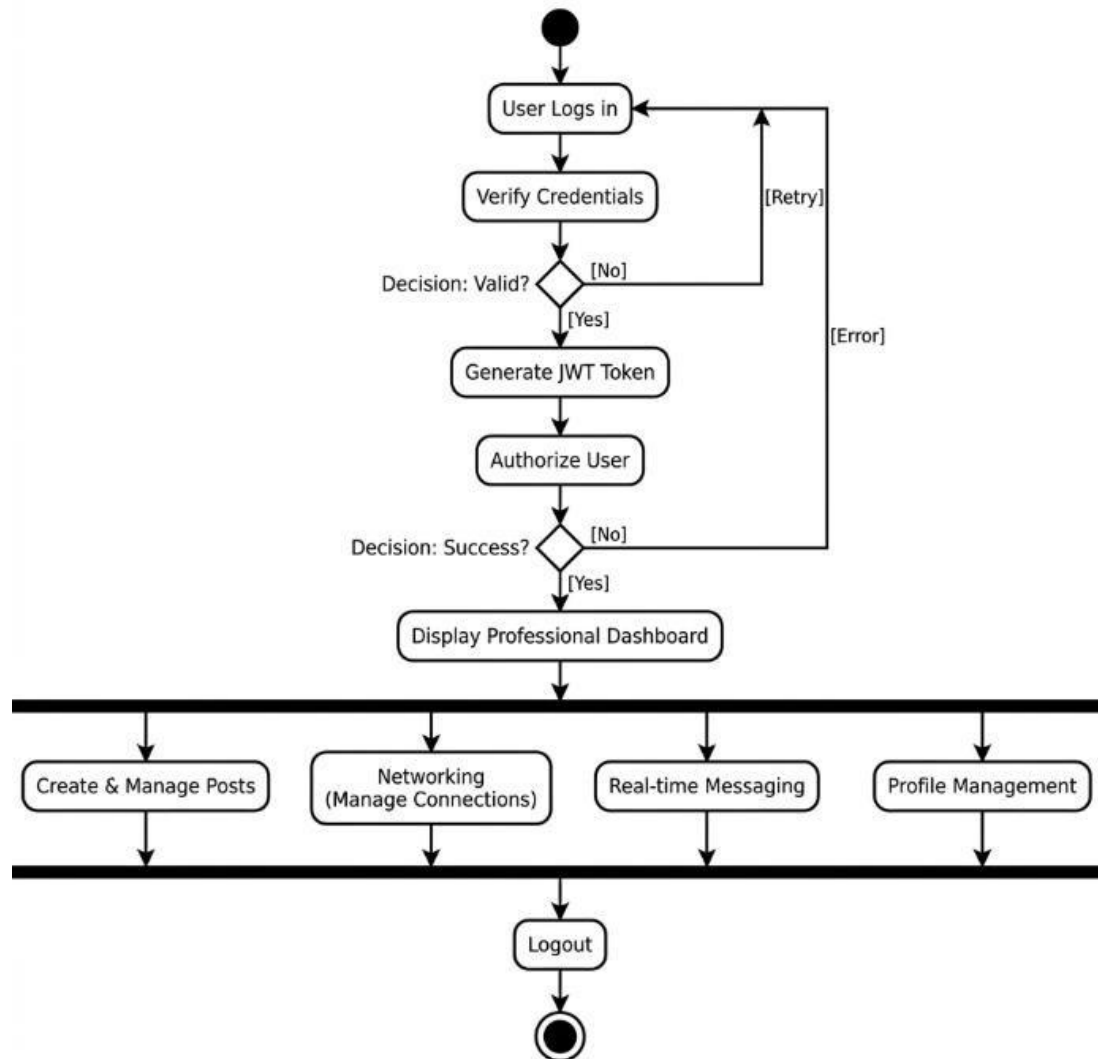
2. Admin Activity Diagram:

UML Activity Diagram: Recruiter Role in Connexus



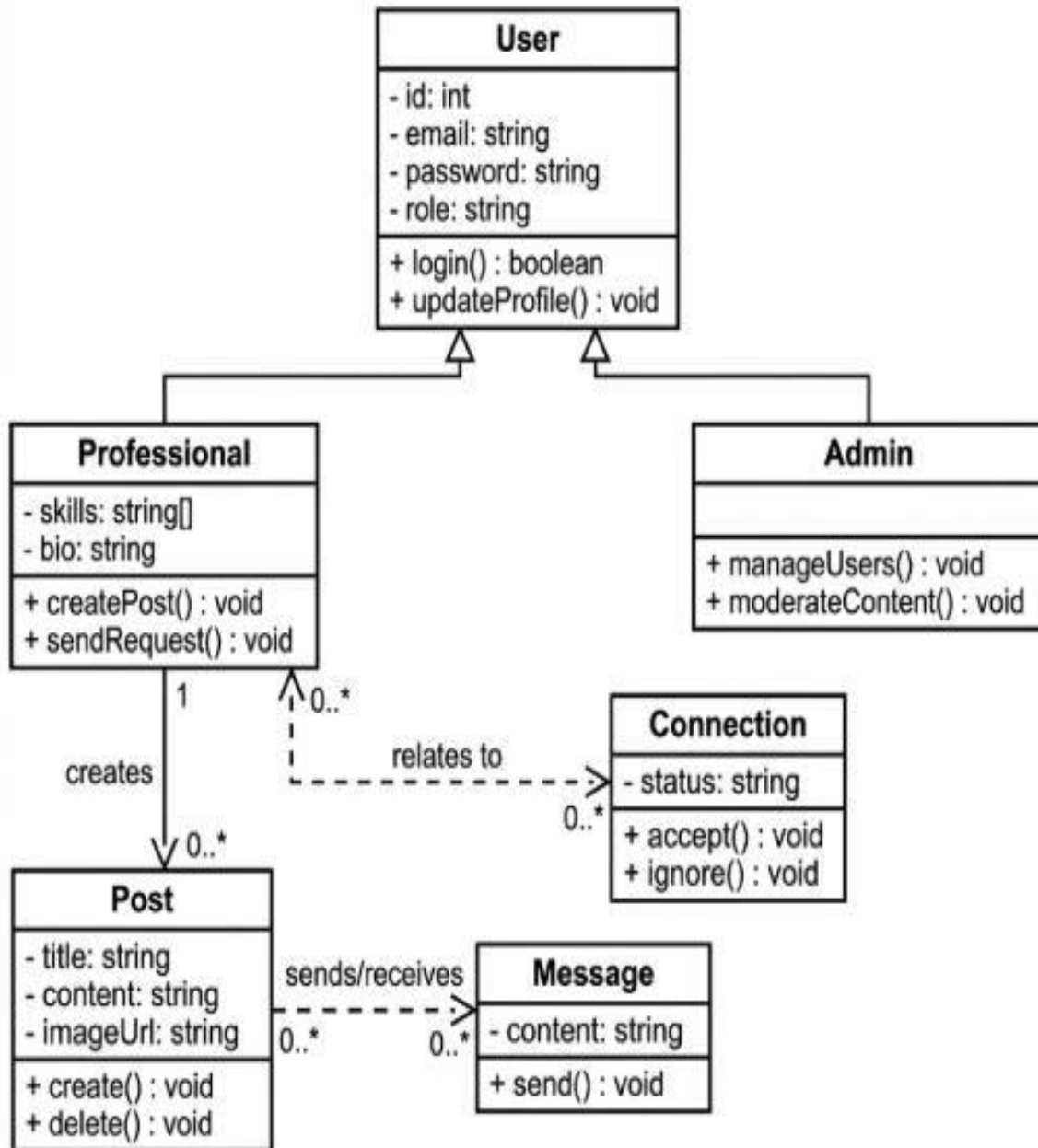
3. User Activity Diagram

UML Activity Diagram - Professional User Role in Connexus



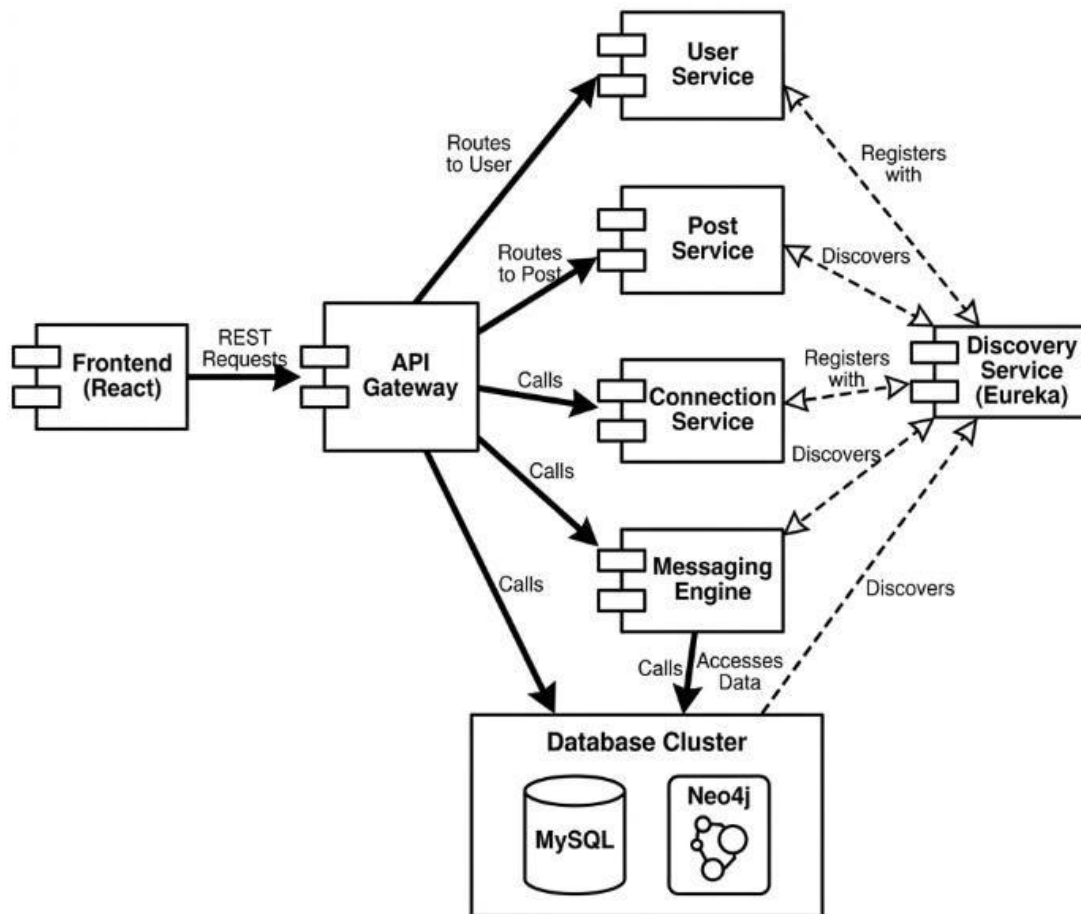
3.5 Class Diagram:

UML Class Diagram: Connexus



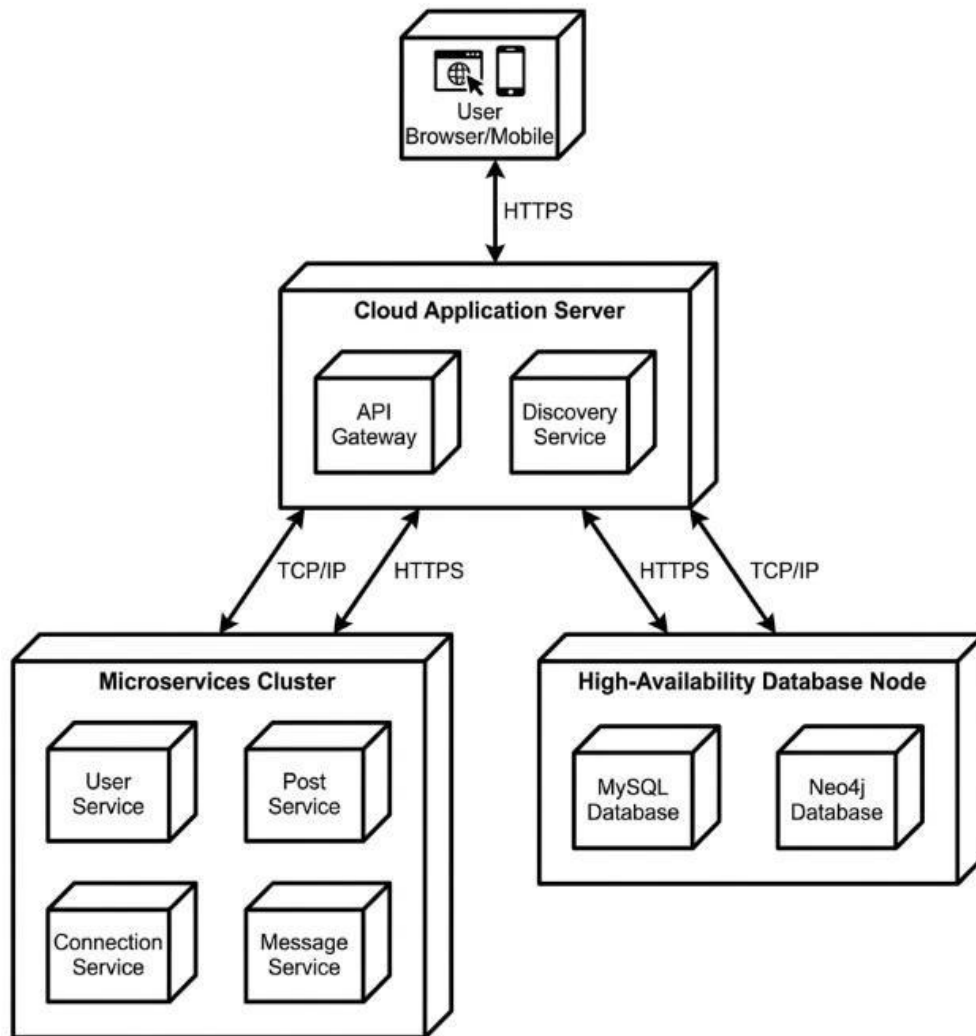
3.6 Component Diagram:

Connexus



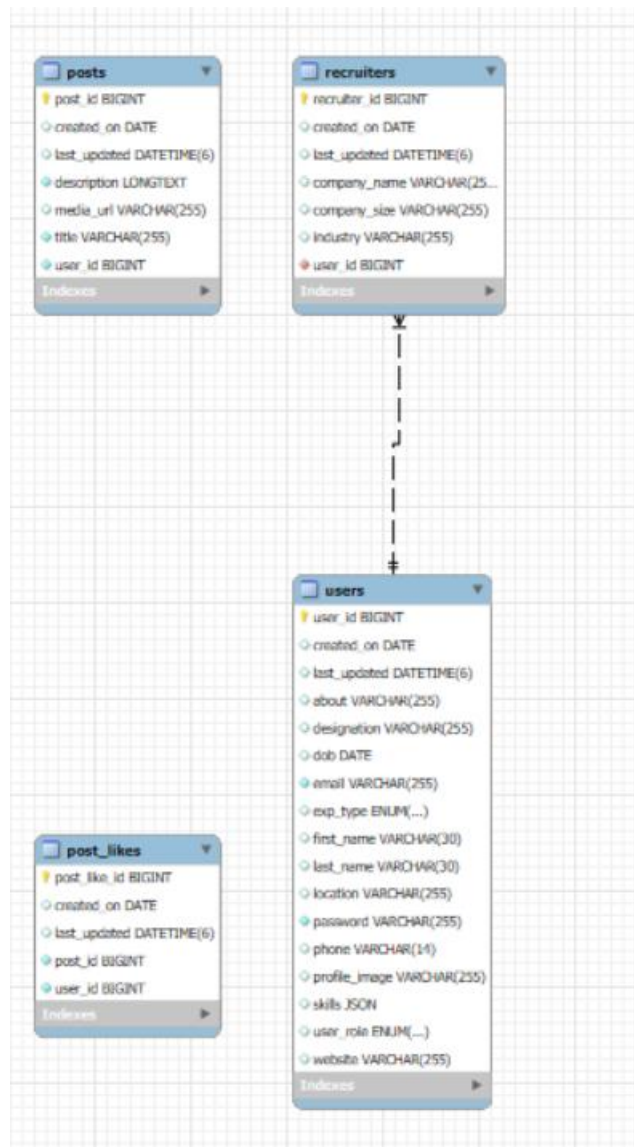
3.7 Sequence Diagram

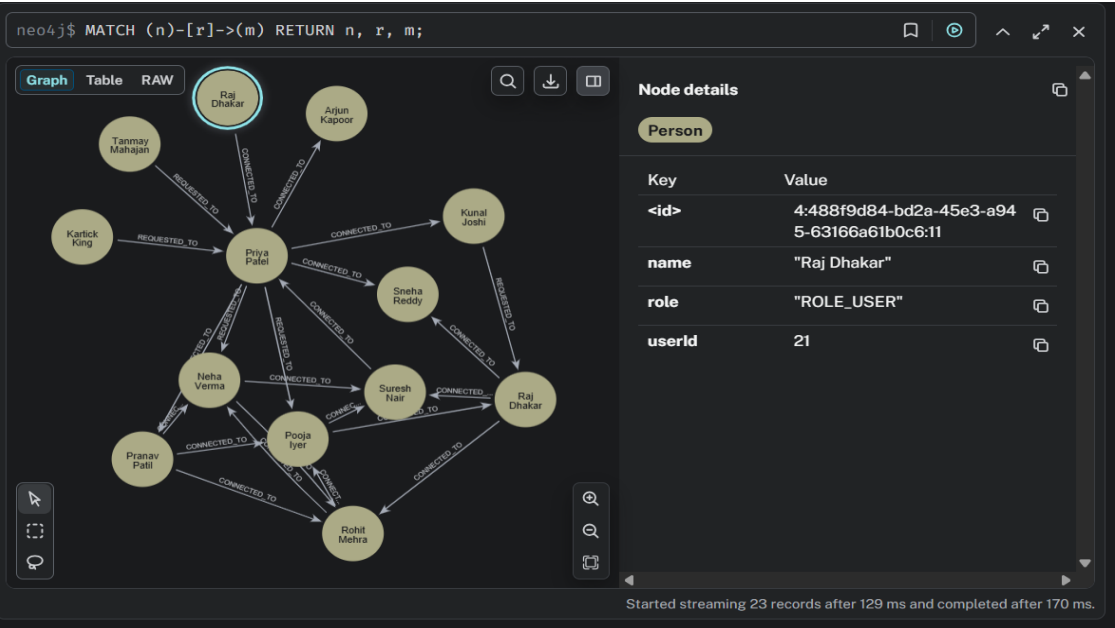
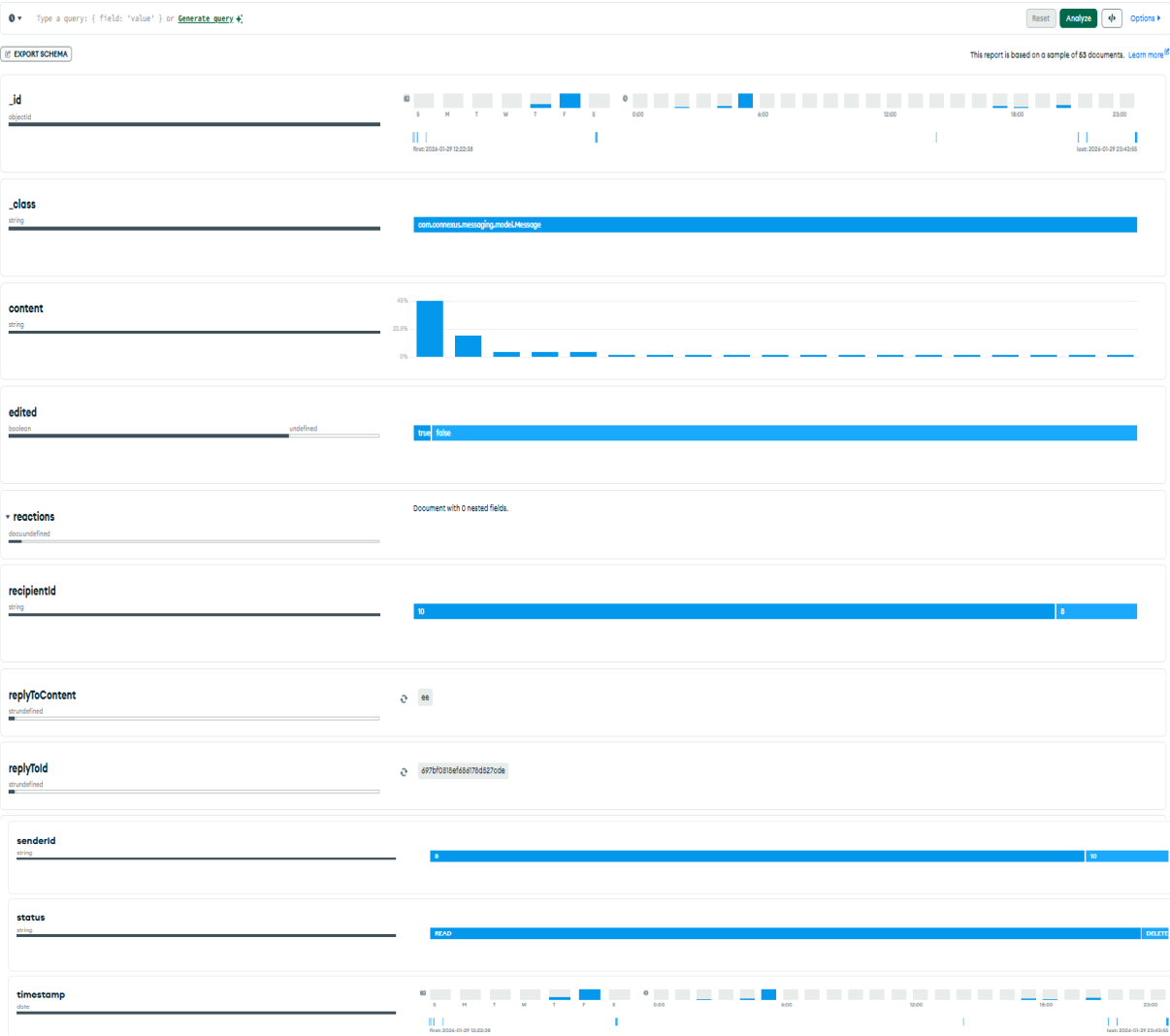
UML Deployment Diagram: Connexus



4. DATABASE DESIGN

4.1 Design:

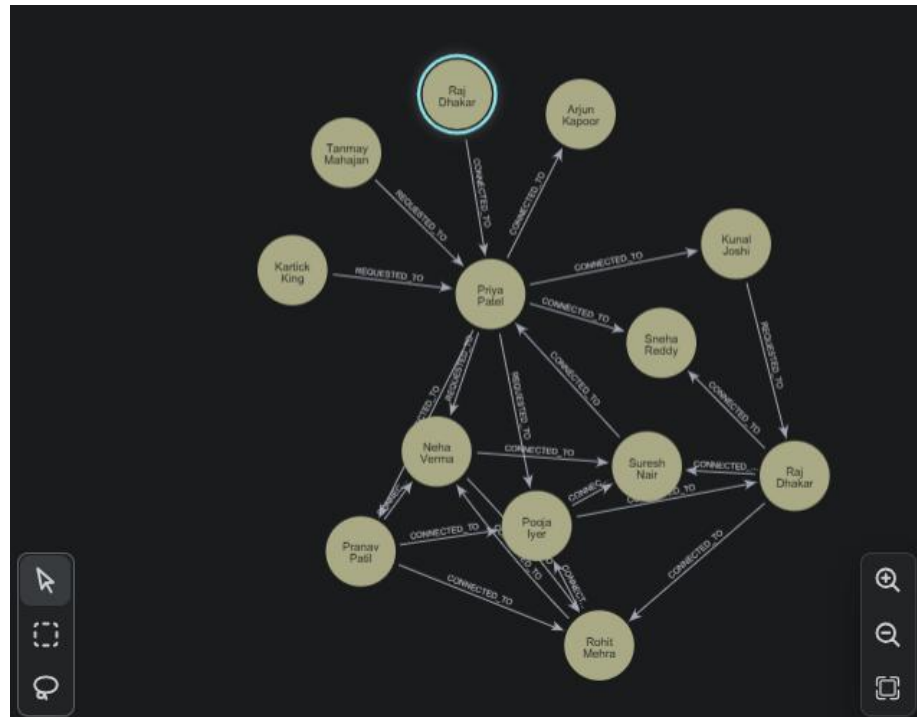




4.2 Tables:

The following table structures depict the database design.

Graph Connection:



User Table:

Database changed mysql> select * from users;														
user_id	created_on	last_updated	about	phone	profile_image	designation	dob	email	exp_type	first_name	last_name	location	skills	
user_role	website													
1	2024-01-10	2026-01-28 22:43:01.239628	Platform administrator with backend experience			Platform Administrator	1990-05-12	p@gmail.com	SENIOR	Pranav	Patil	Kolhapur, Maharashtra, India	["spring-boot", "react", "mysql", "hibernate"]	
2	2024-01-11	2026-01-28 22:37:46.839976	System administrator managing enterprise systems			System Administrator	1988-08-20	neha.admin@gmail.com	MANAGER	Neha	Verma	Delhi, India	["java", "spring-security", "mysql"]	
3	2024-01-12	2026-02-02 12:38:04.518551	Enterprise IT recruiter for devs			Lead Technical Recruiter	1985-03-15	rohit@infosys.com	LEAD	Rohit	Mehra	Bengaluru, Karnataka, India	["java", "spring-boot", "microservices", "docker"]	
4	2024-01-13	2026-01-28 22:37:47.013106	Talent acquisition specialist			Talent Acquisition Specialist	1987-06-25	pooja@tcs.com	MIDLEVEL	Pooja	Iyer	Chennai, Tamil Nadu, India	["reactjs", "javascript", "html", "css"]	
5	2024-01-14	2026-01-28 22:37:47.094780	Senior technical recruiter			Senior Recruiter	1982-09-10	suresh@wipro.com	SENIOR	Suresh	Nair	Hyderabad, Telangana, India	["java", "neo4j", "mysql"]	
6	2024-01-15	2026-01-28 22:37:47.174980	Startup recruiter handling end-to-end hiring			Startup Recruiter	1991-01-19	ananya@startup.com	MIDLEVEL	Ananya	Gupta	Mumbai, Maharashtra, India	["javascript", "reactjs", "node.js"]	
7	2024-01-16	2026-01-28 22:37:47.266627	Backend developer exploring opportunities			Backend Developer	1998-04-12	rahul@gmail.com	JUNIOR	Rahul	Singh	Pune, Maharashtra, India	["java", "spring-boot", "mysql"]	
8	2024-01-17	2026-01-28 22:37:47.598231	Frontend developer specializing in React			Frontend Developer	1997-07-08	priya@gmail.com	MIDLEVEL	Priya	Patel	Ahmedabad, Gujarat, India	["reactjs", "javascript", "html", "css"]	
9	2024-01-18	2026-01-28 22:37:47.428811	Java backend engineer			Java Backend Engineer	1996-11-22	rlrur219@gmail.com	MIDLEVEL	Kunal	Joshi	Jaipur, Rajasthan, India	["java", "spring-boot", "mysql"]	
10	2024-01-19	2026-02-02 14:04:59.568060	Full-stack engineer with cloud exposure			Full Stack Engineer	1995-02-02	sneha@gmail.com	SENIOR	Sneha	Reddy	Bengaluru, Karnataka, India	["java", "reactjs", "docker"]	
11	2024-01-20	2026-01-28 22:37:47.672684	Business analyst transitioning to tech			Business Analyst	1994-10-09	divya@gmail.com	MIDLEVEL	Divya	Malhotra	Noida, Uttar Pradesh, India	["sql", "mysql", "excel"]	
12	2024-01-21	2026-01-28 22:37:47.741705	Backend engineer with API experience			Senior Backend Engineer	1993-03-30	vikas@gmail.com	SENIOR	Vikas	Yadav	Gurgaon, Haryana, India	["java", "spring-boot", "mysql"]	
13	2024-01-22	2026-01-28 22:37:47.822185	QA engineer learning automation			QA Engineer	1992-06-18	ritu@gmail.com	MIDLEVEL	Ritu	Chopra	Chandigarh, India	["selenium", "java", "testing"]	
14	2024-01-23	2026-01-28 22:37:47.905975	DevOps engineer			DevOps Engineer	1991-08-27	manish@gmail.com	SENIOR	Manish	Agarwal	Bengaluru, Karnataka, India	["docker", "kubernetes", "aws"]	
15	2024-01-24	2026-01-28 22:37:47.988599	Database specialist			Database Engineer	1990-09-05	pankaj@gmail.com	SENIOR	Pankaj	Mishra	Lucknow, Uttar Pradesh, India	["mysql", "postgresql"]	

Message Collection:

```

TypeError: db.messages.findAll is not a function
connexus_chat> db.messages.find();
[
  {
    _id: ObjectId('698033b9a7bc063407b98ba8'),
    senderId: '8',
    recipientId: '10',
    content: 'hi',
    timestamp: ISODate('2026-02-02T05:18:49.052Z'),
    status: 'READ',
    edited: false,
    reactions: {},
    _class: 'com.connexus.messaging.model.Message'
  },
  {
    _id: ObjectId('698033d6a7bc063407b98ba9'),
    senderId: '8',
    recipientId: '10',
    content: 'hello',
    timestamp: ISODate('2026-02-02T05:19:18.183Z'),
    status: 'READ',
    edited: false,
    reactions: {},
    _class: 'com.connexus.messaging.model.Message'
  },
  {
    _id: ObjectId('698033dda7bc063407b98baa'),
    senderId: '8',
    recipientId: '10',
    content: 'how are you?',
    timestamp: ISODate('2026-02-02T05:19:25.439Z'),
    status: 'READ',
    edited: false,
    reactions: { '10': '😬' },
    _class: 'com.connexus.messaging.model.Message'
  },
  {
    _id: ObjectId('698035f7a7bc063407b98bab'),
    senderId: '10',
    recipientId: '8',
    content: 'I am fine.',
    timestamp: ISODate('2026-02-02T05:28:23.662Z'),
    status: 'READ',
    edited: false,
    reactions: {},
    _class: 'com.connexus.messaging.model.Message'
  },
  {
    _id: ObjectId('69803619a7bc063407b98bac'),

```

Recruiter Table:

```
mysql> select * from recruiters;
```

recruiter_id	created_on	last_updated	company_name	company_size	industry	user_id
1	2024-01-12	2026-01-28 22:34:07.000000	Infosys	100,000+	IT Services	3
2	2024-01-13	2026-01-28 22:34:07.000000	Tata Consultancy Services	100,000+	IT Consulting	4
3	2024-01-14	2026-01-28 22:34:07.000000	Wipro	100,000+	Technology	5
4	2024-01-15	2026-01-28 22:34:07.000000	NextGen Startup	50-200	Product-Based	6

```
4 rows in set (0.01 sec)
```

Posts Table:

```
mysql> select * from recruiters;
```

recruiter_id	created_on	last_updated	company_name	company_size	industry	user_id
1	2024-01-12	2026-01-28 22:34:07.000000	Infosys	100,000+	IT Services	3
2	2024-01-13	2026-01-28 22:34:07.000000	Tata Consultancy Services	100,000+	IT Consulting	4
3	2024-01-14	2026-01-28 22:34:07.000000	Wipro	100,000+	Technology	5
4	2024-01-15	2026-01-28 22:34:07.000000	NextGen Startup	50-200	Product-Based	6

```
4 rows in set (0.01 sec)
```

PostLike Table

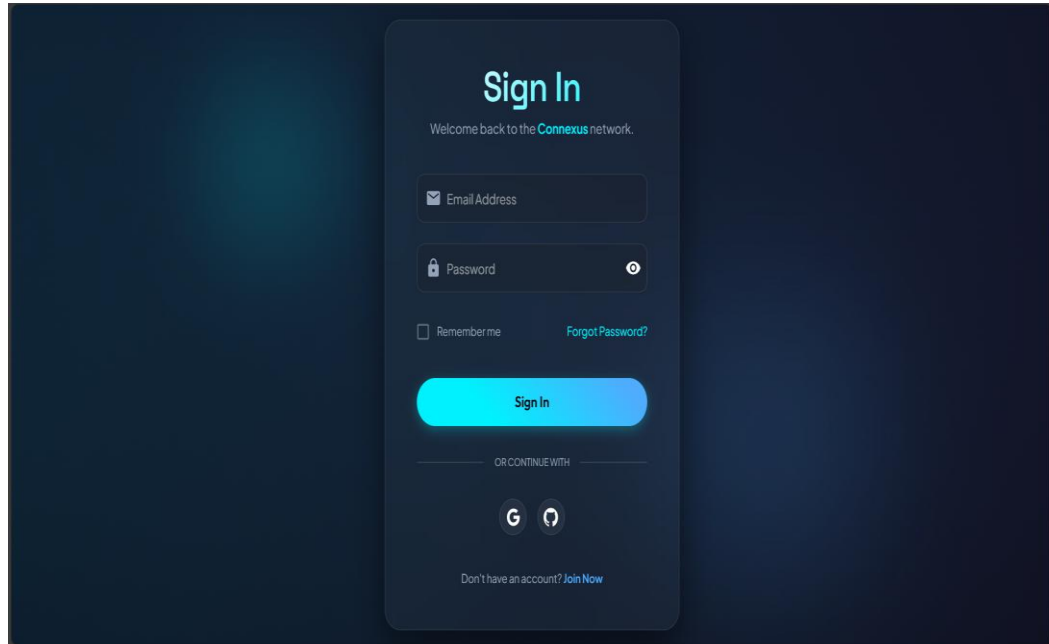
```
mysql> select * from post_likes;
```

post_like_id	created_on	last_updated	post_id	user_id
1	2024-01-05	2024-01-05 10:15:30.000000	101	1
2	2024-01-05	2024-01-05 10:18:12.000000	101	2
3	2024-01-06	2024-01-06 09:05:44.000000	102	1
4	2024-01-06	2024-01-06 09:10:01.000000	102	3
5	2024-01-07	2024-01-07 14:45:20.000000	103	2
6	2024-01-07	2024-01-07 15:01:55.000000	103	4
7	2024-01-08	2024-01-08 08:30:00.000000	104	1
8	2024-01-08	2024-01-08 08:31:10.000000	104	3

```
8 rows in set (0.00 sec)
```

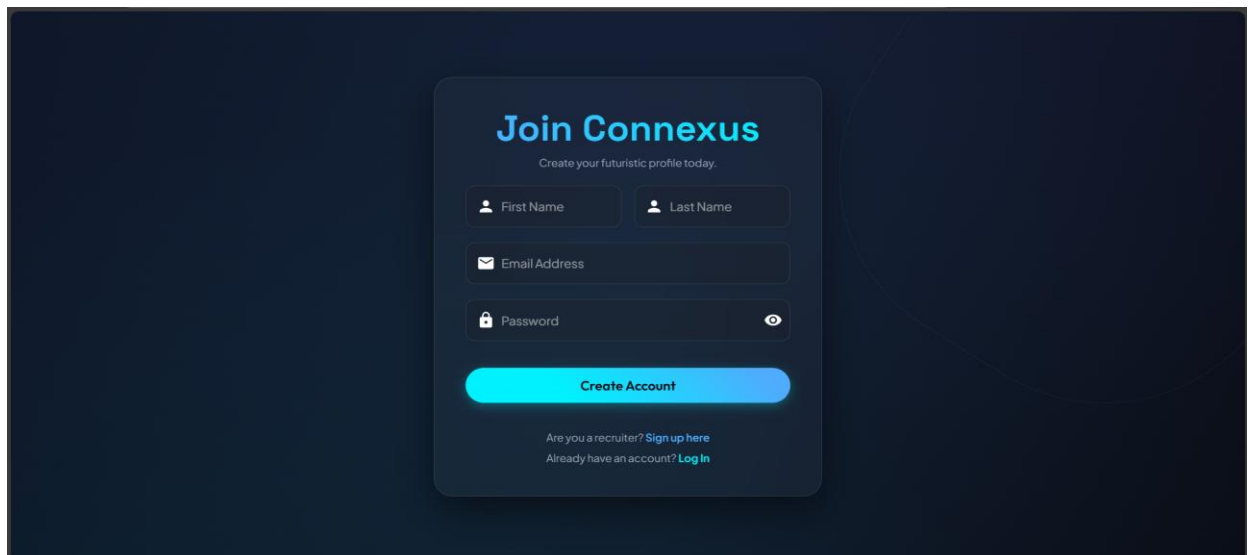

5. SNAPSHOTS

SignIn Page:



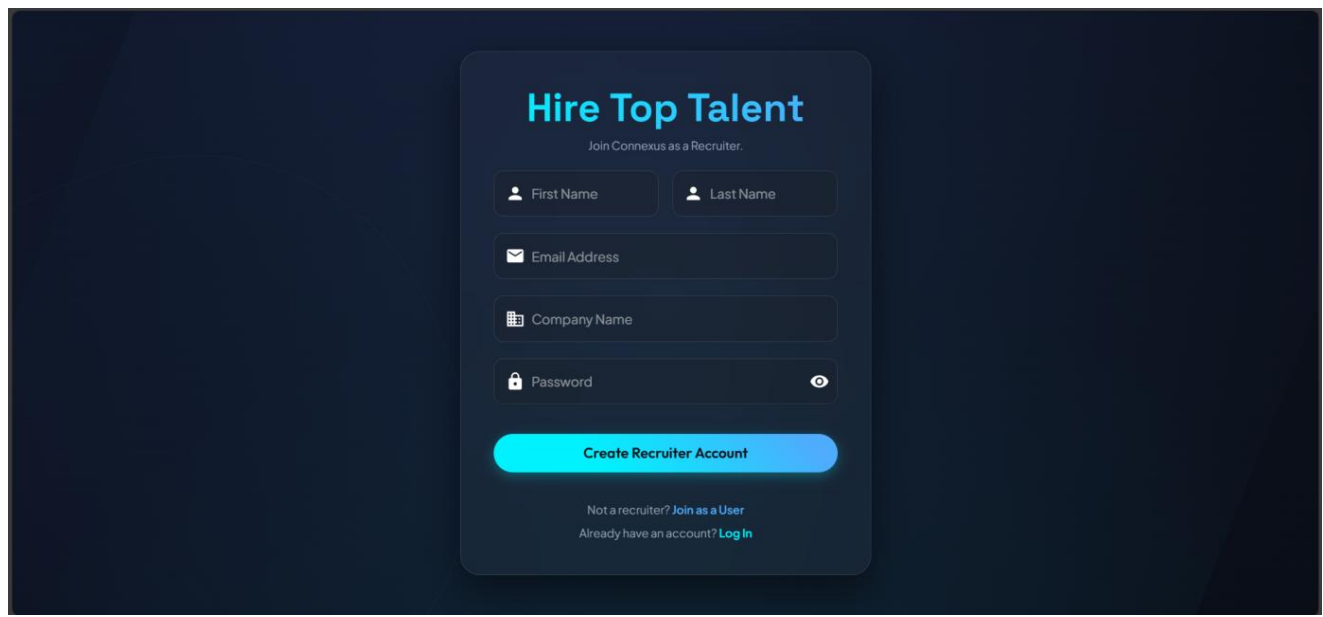
The image shows a 'Sign In' page for the Connexus network. The page has a dark blue background with a central white card. The card contains the title 'Sign In' in a large, bold, dark blue font. Below the title is a subtitle 'Welcome back to the Connexus network.' in a smaller, lighter blue font. The card features two input fields: 'Email Address' with an envelope icon and 'Password' with a lock icon and a toggle eye icon. Below these fields are two links: 'Remember me' with a checkbox and 'Forgot Password?' in a lighter blue font. A large, rounded, light blue button with the text 'Sign In' in dark blue is positioned below the links. Below the button is a separator line with the text 'OR CONTINUE WITH' in the center. Underneath the separator are two circular icons: a Google 'G' logo and a GitHub logo. At the bottom of the card, there is a link 'Don't have an account? Join Now' in a lighter blue font.

User SignUp Page:



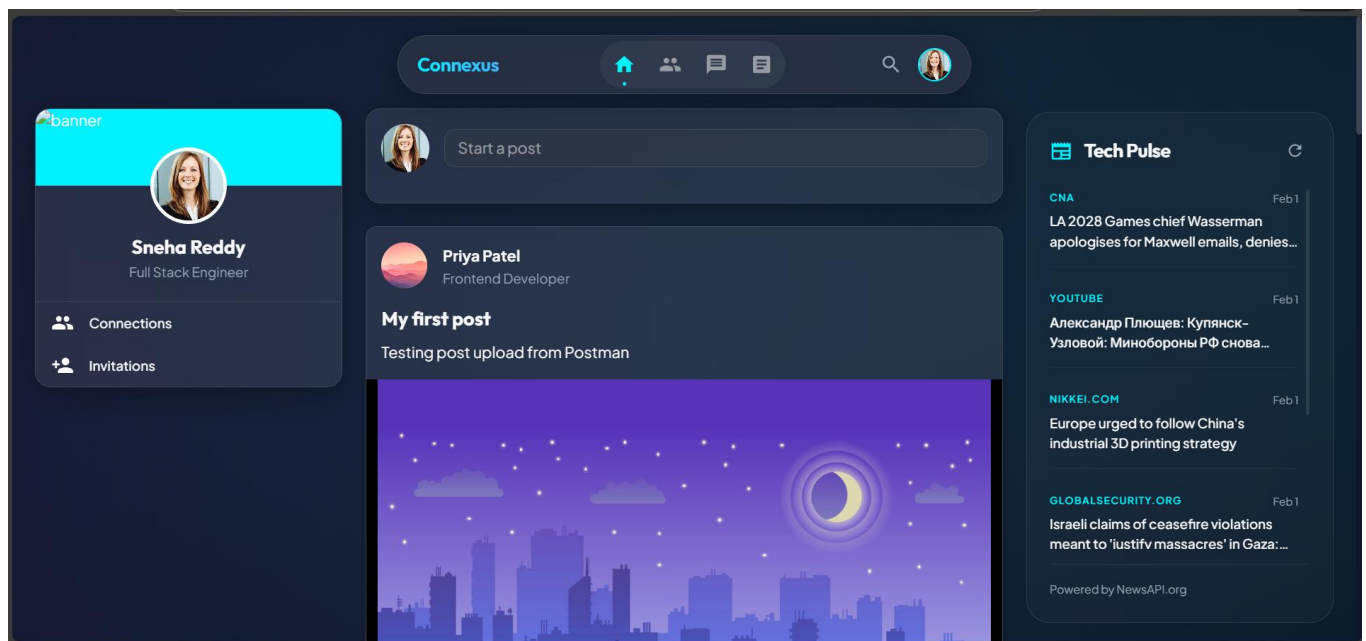
The image shows a 'Join Connexus' page for creating a new account. The page has a dark blue background with a central white card. The card contains the title 'Join Connexus' in a large, bold, dark blue font. Below the title is a subtitle 'Create your futuristic profile today.' in a smaller, lighter blue font. The card features four input fields: 'First Name' and 'Last Name' (each with a person icon), 'Email Address' (with an envelope icon), and 'Password' (with a lock icon and a toggle eye icon). A large, rounded, light blue button with the text 'Create Account' in dark blue is positioned below the input fields. Below the button, there are two links: 'Are you a recruiter? Sign up here' and 'Already have an account? Log In', both in a lighter blue font.

Recruiter SignUp Page:



The Recruiter SignUp Page features a dark blue background with a central white card. The card has the heading "Hire Top Talent" in blue, followed by the subtext "Join Connexus as a Recruiter." Below this are five input fields: "First Name", "Last Name", "Email Address", "Company Name", and "Password" (with a toggle eye icon). A prominent blue button labeled "Create Recruiter Account" is positioned below the fields. At the bottom of the card, there are two links: "Not a recruiter? Join as a User" and "Already have an account? Log In".

Home Page:




Profile Page:

×

Profile

✎ Edit Profile



Sneha Reddy ✓
@Sneha Reddy
Full Stack Engineer

Intro ✎
Born 1995-02-02
Bengaluru, Karnataka, India
<https://sneha-fullstack.dev>
SENIOR

About ✎
Full-stack engineer with cloud exposure

Contact Information ✎
Email Address: sneha@gmail.com
Phone Number: 919977889900

Skills ✎
java reactjs docker


Activity
Posted an update
Interview Preparation Tips

Profile Edit Page:

×

Profile

Save Cancel



First Name
Sneha

Last Name
Reddy

Designation
Full Stack Engineer

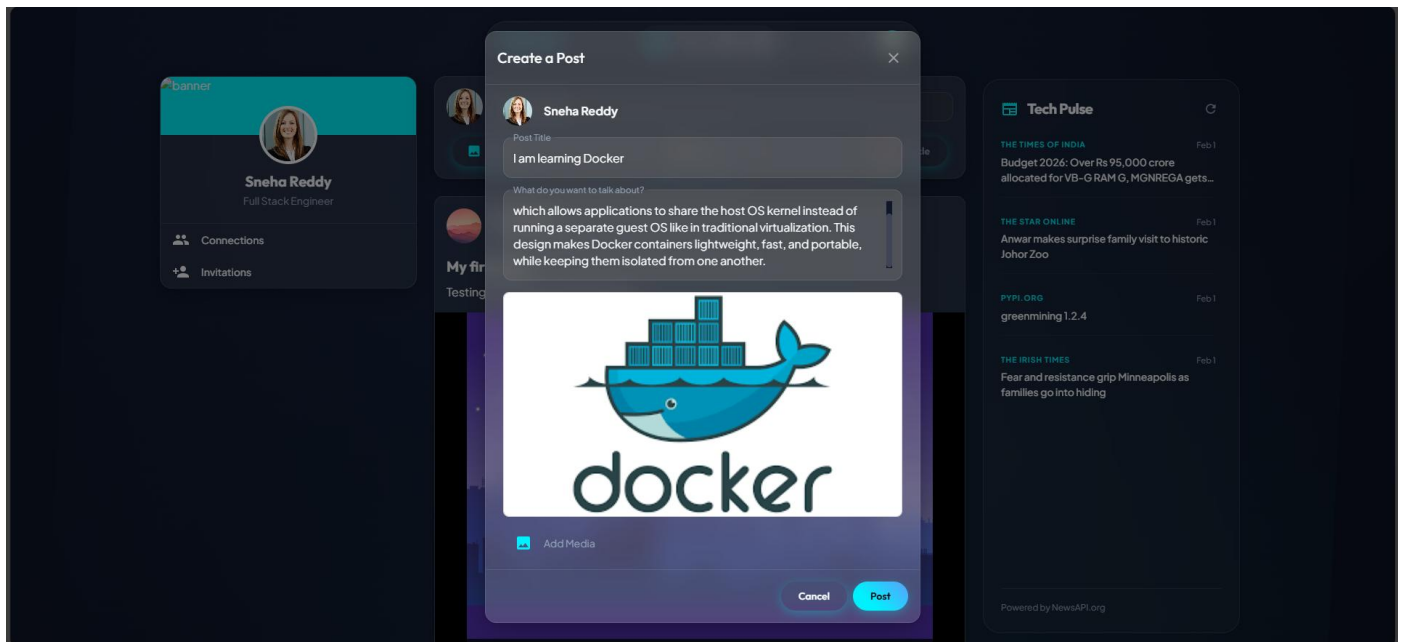
Intro
Location
Bengaluru, Karnat...
Website
<https://sneha-fullstc>
Experience Level
SENIOR

About
Full-stack engineer with cloud exposure

Contact Information
Email Address: sneha@gmail.com
Phone Number: 919977889900

Skills
Skills
java reactjs docker Add skills (e.g. React, Java)

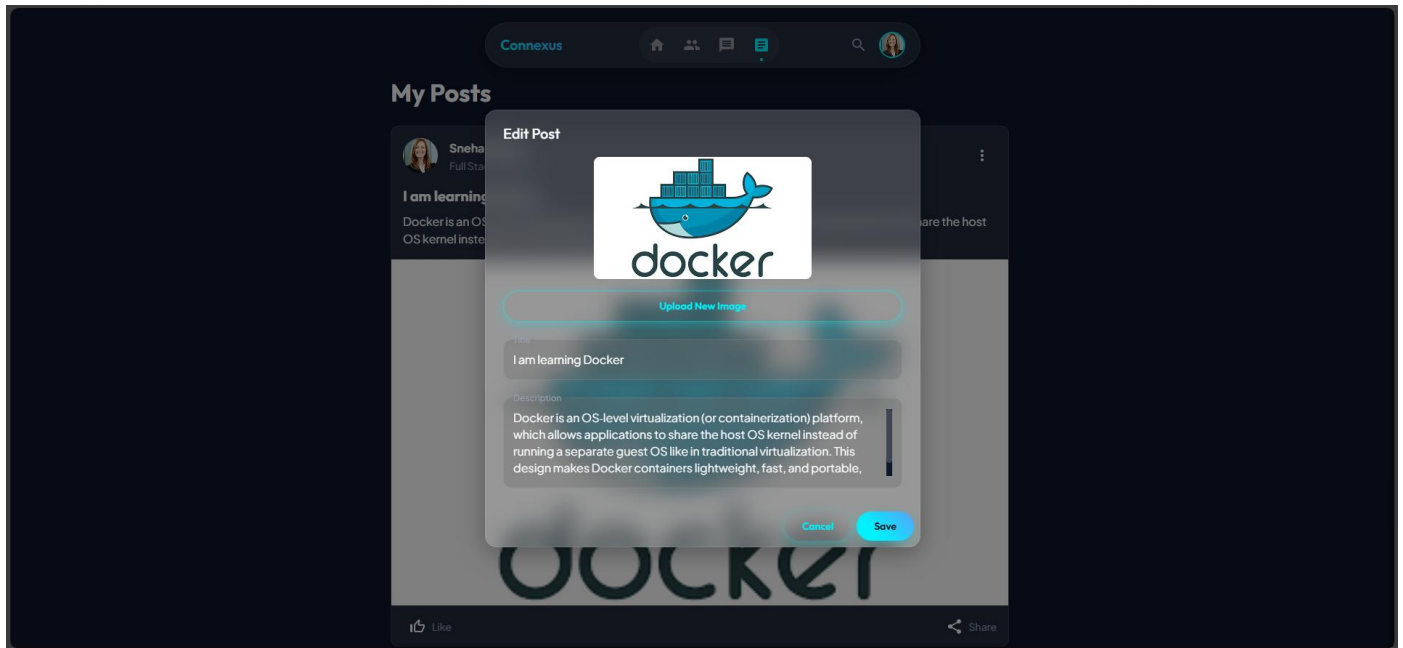
Upload Post Image Page:



All Post Page:



Post Edit Page:



Invitations Page:



All Connections Page:

People you may know

P

Pranav Patil
ROLE_USER

Connect

N

Neha Verma
ROLE_USER

Connect

R

Rohit Mehra
ROLE_RECRUITER

Connect

P

Pooja Iyer
ROLE_RECRUITER

Connect

S

Suresh Nair
ROLE_RECRUITER

Connect





R



Raj Dhakar
ROLE_USER

Connect


Network Page:

Connexus






My Network



Priya Patel
ROLE_USER

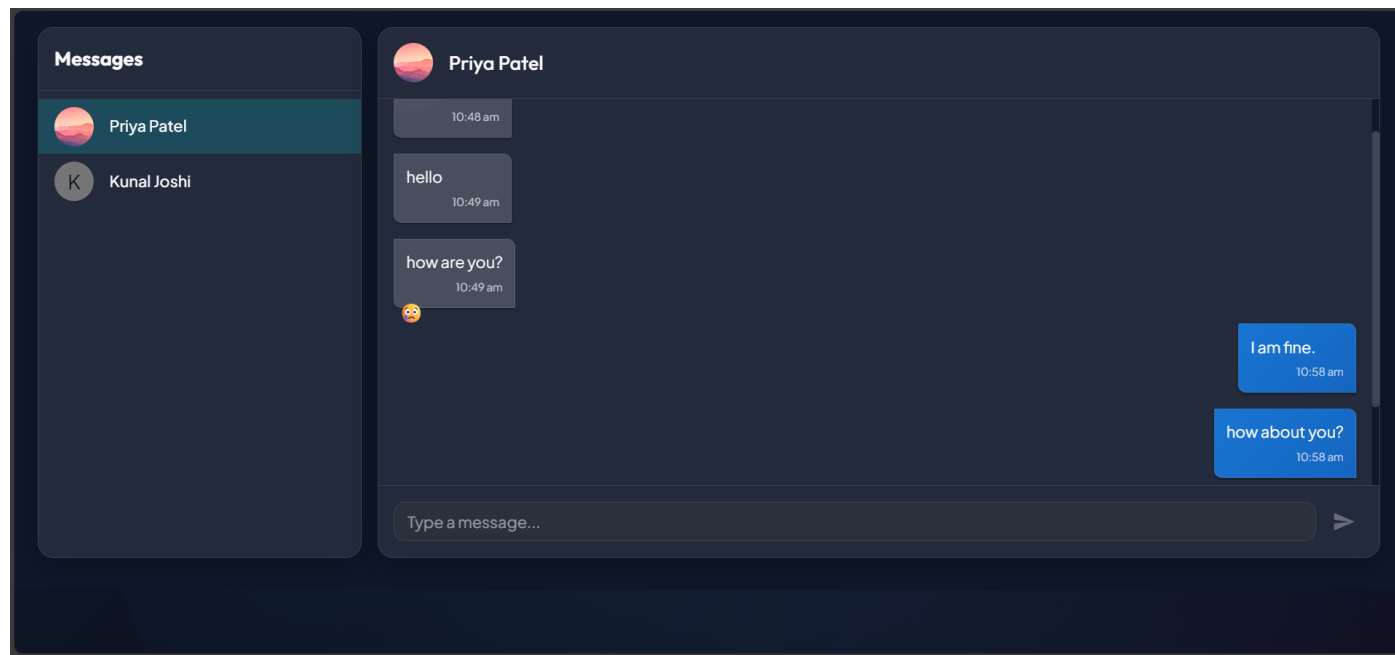
Message



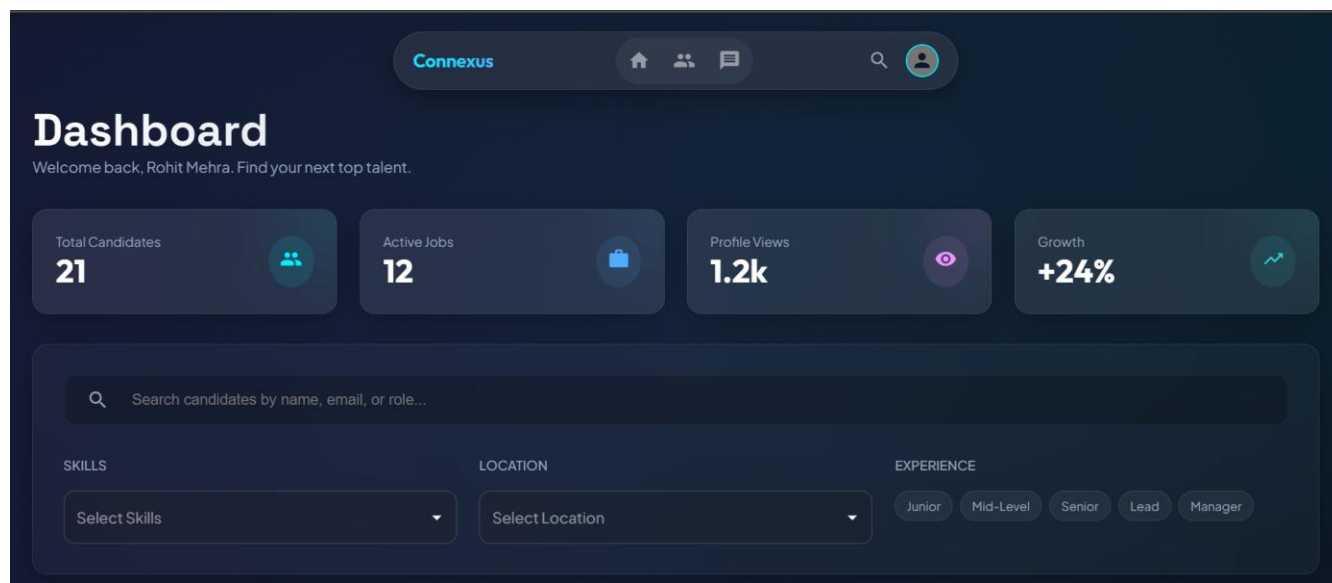
Kunal Joshi
ROLE_USER

Message
















Message Page:



Recruiter DashBoard Page:



Recruiter User List:

Recent Candidates				Showing 21 candidates	
Candidate		Connexus		Action	
	Pranav Patil	p@gmail.com spring-boot react mysql hibernate Kolhapur, Maharashtra, India	Candidate		
	Neha Verma	neha.admin@gmail.com java spring-security mysql Delhi, India	Candidate		
	Rohit Mehra	rohit@infosys.com java spring-boot microservices docker Bengaluru, Karnataka, India	Recruiter		
	Pooja Iyer	pooja@tcs.com css html reactjs javascript Chennai, Tamil Nadu, India	Recruiter		
	Suresh Nair	suresh@wipro.com java neo4j mysql Hyderabad, Telangana, India	Recruiter		

Recruiter SignUp Page:

Hire Top Talent

Join Connexus as a Recruiter.

Create Recruiter Account

Not a recruiter? [Join as a User](#)

Already have an account? [Log In](#)


Profile Edit Page:

×

Profile

Save

Cancel



First Name

Sneha

Last Name

Reddy

Designation

Full Stack Engineer

Intro

Location

Bengaluru, Karnat...

Website

https://sneha-fullst...

Experience Level

SENIOR

About

Full-stack engineer with cloud exposure

Contact Information

Email Address

sneha@gmail.com

Phone Number

919977889900

Skills

java

reactjs

docker

Add skills (e.g. React, Java)

6. **CONCLUSION**

In conclusion, Connexus addresses key requirements for modern professional networking and recruitment systems, providing a solid foundation for further development and improvement. Its modular microservices architecture and adherence to high-security standards and real-time communication protocols position it well for continued success and adaptation in the evolving professional networking landscape.

The Connexus project exemplifies a sophisticated approach to professional social platforms by leveraging cutting-edge technologies and methodologies. This application is designed to meet the needs of both professionals and recruiters, ensuring a seamless experience for networking, content sharing, and candidate discovery while adhering to stringent security standards and performance metrics.

Its blend of modern technologies, robust security, and user-centric design positions it as a valuable tool for users seeking to enhance their networking and talent acquisition capabilities. The project's architecture and features provide a strong foundation for ongoing development, making it adaptable to future technological advancements and market trends in the professional industry.

7. REFERENCES

1. <https://docs.spring.io/spring-security/reference/>
2. <https://docs.spring.io/spring-boot/index.html/>
3. Garcia-Molina, H., Ullman, J. D., & Widom, J. (2008). *Database Systems: The Complete Book*. Prentice Hall.
4. Codd, E. F. (1970). *A Relational Model of Data for Large Shared Data Banks*. Communications of the ACM, 13(6), 377-387.
5. Fowler, M. (2003). *Patterns of Enterprise Application Architecture*. Addison- Wesley.
6. Evans, E. (2003). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley.
7. Soni, D., Nord, R. L., & Hofmeister, C. (1995). *Software Architecture in Industrial Applications*. Proceedings of the 17th International Conference on Software Engineering, 196-207.