

I. HOSTING APPLICATION COMPONENTS

A. Host Site - Microservices Deployment

Microservice	Purpose	Host Platform	Specs
User Service	Authentication, user management	AWS EC2 / Docker	t3.small, Port 8081
Query Service	Query handling, AI processing	AWS EC2 / Docker	t3.medium, Port 8082
Admin Service	Admin operations, analytics	AWS EC2 / Docker	t3.small, Port 8083
API Gateway	Request routing, load balancing	AWS API Gateway	Entry point
User Database	User data	AWS RDS MySQL	db.t3.micro, 10GB
Query Database	Query data	AWS RDS MySQL	db.t3.micro, 10GB
Frontend	UI	AWS S3 + CloudFront	Static hosting

B. Deployment Strategy

Step 1: Microservices Setup

1.1 User Service (Microservice 1)

- **Port:** 8081
- **Database:** users_db
- **Functions:** Register, login, profile management
- **Deploy:** Docker container on EC2

1.2 Query Service (Microservice 2)

- **Port:** 8082
- **Database:** queries_db
- **Functions:** Submit query, AI classification, response generation
- **Deploy:** Docker container on EC2

1.3 Admin Service (Microservice 3)

- **Port:** 8083
- **Database:** Shared access to both DBs
- **Functions:** User management, query analytics, dashboard
- **Deploy:** Docker container on EC2

Step 2: API Gateway Configuration

AWS API Gateway Routes:

/api/users/** → User Service (8081)

/api/queries/** → Query Service (8082)

/api/admin/** → Admin Service (8083)

Step 3: Service Discovery

Using Eureka Server (Netflix OSS)

Eureka Server: <http://eureka-server:8761>

Registered Services:

- user-service → <http://10.0.1.10:8081>

- query-service → <http://10.0.1.11:8082>

- admin-service → <http://10.0.1.12:8083>

Step 4: Inter-Service Communication

REST API Calls Between Services:

Query Service → User Service:

Admin Service → Both Services:

Step 5: Database Setup

Separate Databases per Service:

User Database (users_db)

Query Database (queries_db)

Step 6: Frontend Deployment

- Deploy to S3 + CloudFront
 - Configure to call API Gateway
 - Update API URLs to point to gateway
-

C. Security Measures

1. Service-Level Security

Each Microservice:

- BCrypt password encryption (User Service)
- JWT token authentication (all services)
- Input validation
- Rate limiting

2. API Gateway Security

- SSL/TLS termination
- Request throttling (1000 req/min per user)
- API key validation
- CORS configuration
- DDoS protection (AWS WAF)

II. USER ACCESS & SYSTEM INTERACTION

A. How End Users Access Services (5 Marks)

Access Methods:

1. Web Browser

- URL: <https://www.assistx.com>
- Requests go through API Gateway
- Gateway routes to appropriate microservice

2. Mobile App

- Base URL: <https://api.assistx.com>
- JWT authentication
- JSON responses

3. Admin Portal

- URL: <https://admin.assistx.com>
- Routes to Admin Service through gateway

B. Pictorial Representation - Microservices Architecture

Diagram 1: Microservices Architecture Overview

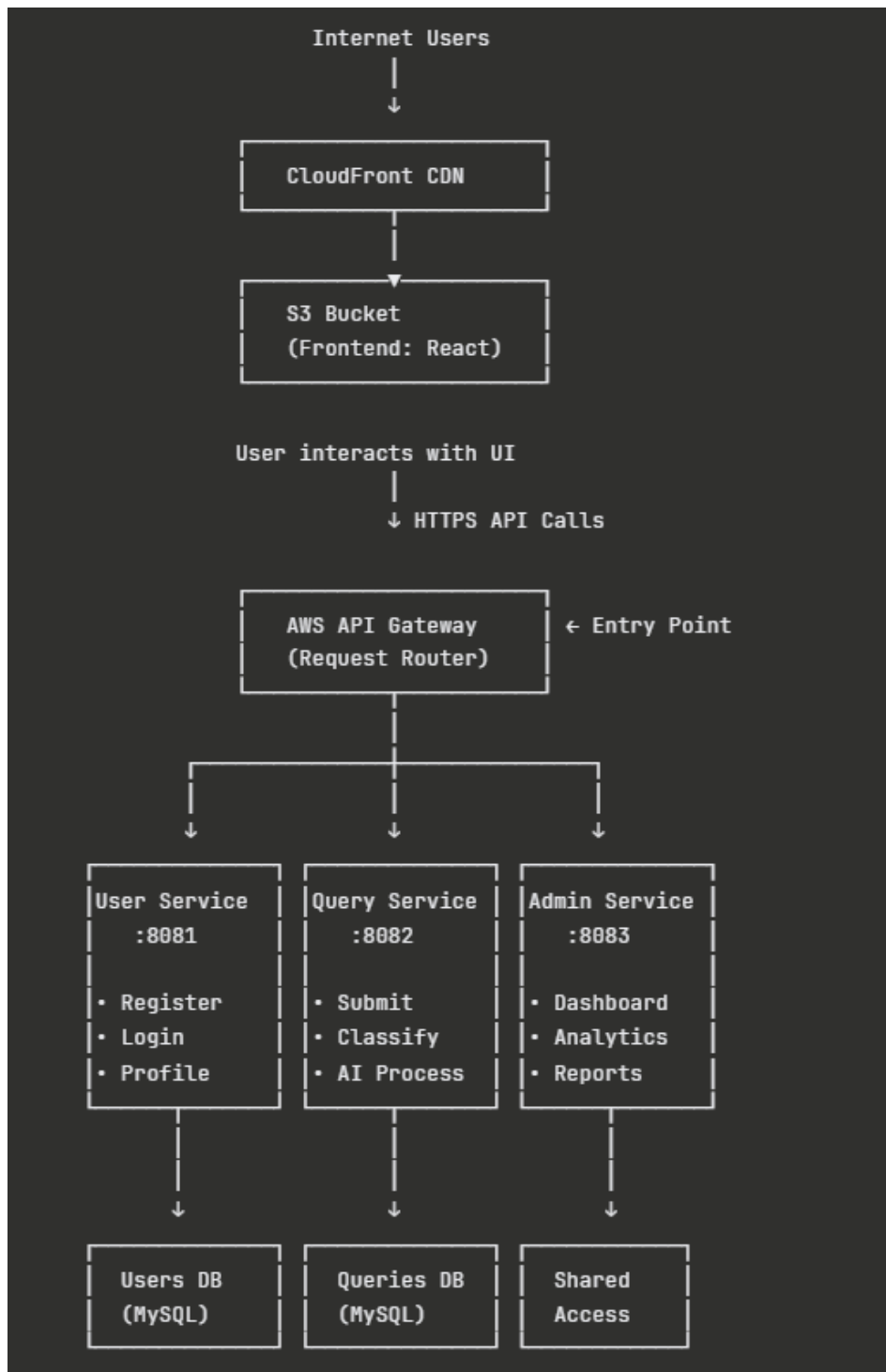


Diagram 2: Microservices Communication

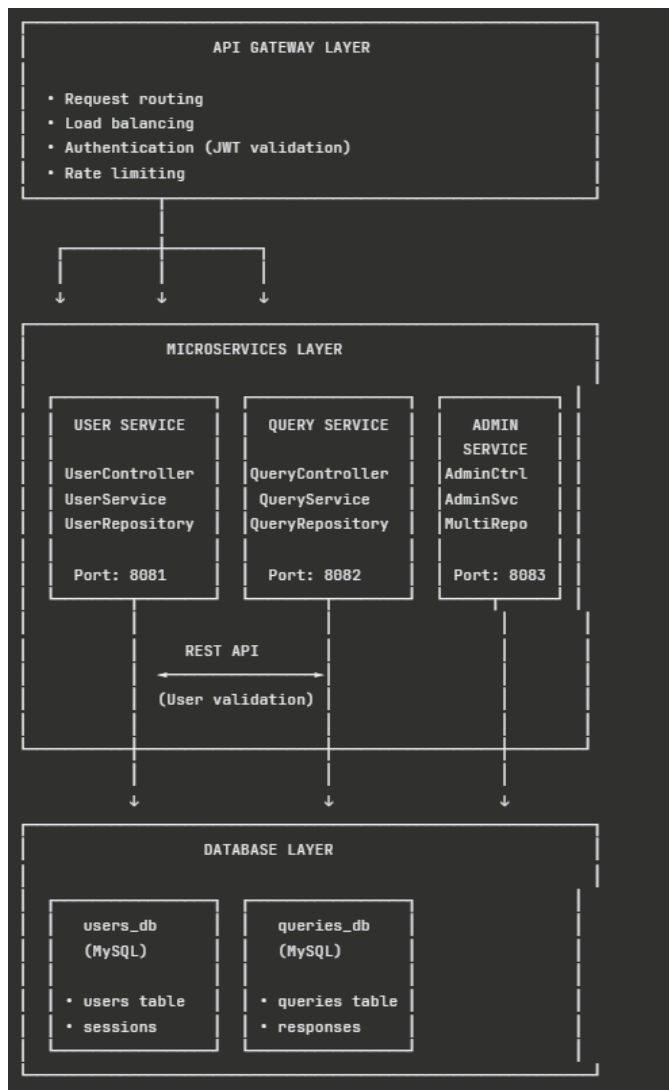


Diagram 3: User Registration Flow (Microservices)

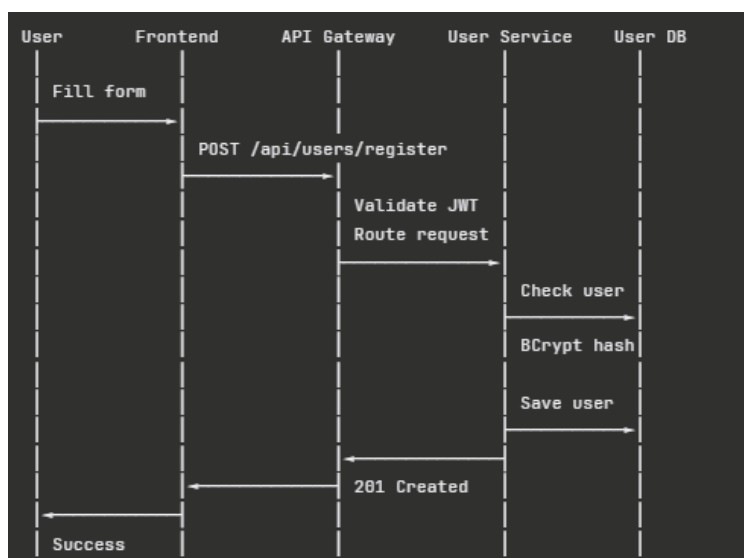


Diagram 4: Query Submission (Inter-Service Communication)

