Project Elements

A) Frontend Applications

1. Customer App / Web

- Browse services (plumbers, electricians, mechanics, etc.)
- Submit service requests
- Track status updates
- Rate & review providers

2. Service Provider App / Web

- Manage profile and availability
- Receive service requests
- Accept/decline jobs
- Communicate with customers
- Track job history

B) Backend Services

1. API Service (Monolith)

Handles:

- User accounts

- Provider profiles
- Service categories
- Service requests lifecycle (pending → accepted → completed)
- Reviews & ratings
- Search endpoints

2. Chat / Real-Time Service

- WebSockets for real-time chat
- Real-time updates on tracking service provider

3. Notification Service

- Asynchronous tasks for:
    - Push notifications
    - Email

4. Authentication Service (or module)

- User authentication (customers & providers)
- Session tokens / JWT

C) Databases

1. Relational Database (PostgreSQL)

- Users
- Service providers

- Service categories
- Requests
- Reviews

2. Firebase Storage / S3

- Images: individual services / service categories / profile pictures
- Videos: presentation of a service

3. Redis Cache (Not necessary for now)

- Caching service search results
- Rate-limiting counters

D) Supporting Infrastructure

- Cloud Load Balancer - Distribute traffic across multiple backend instances (horizontal scaling)
- Content Delivery Network (CDN) - Store and deliver static files (images/videos) from servers located close to the user
- Message Queue - system for handling asynchronous tasks (notifications /emails / analytics)
- Logging & Metrics - track what is happening inside your system (Errors/ Warnings / Latency)

Architectural Patterns ?

A) Modular Monolith

- Faster development
- Easier Management
- Perfect for early-stage products / Keeps everything simple

Modules:

- Auth
- Providers
- Service Requests
- Reviews
- Chat
- Notifications

How will the pieces communicate with each other?

A) Synchronous Communication

GraphQL

- Customer → Backend Server
- Provider → Backend Server
- Backend modules calling each other internally

B) Real-Time Communication

WebSockets (or SSE)

- Chat between customer and provider
- Live updates on service provider location

C) Asynchronous Communication

Message Queue (Pub/Sub)

- Push notifications
- Email messages
- Analytics events
- Provider alerts

AuthN and AuthZ

A) Authentication (AuthN)

1. Customer/Provider Login

- Email/password or OAuth
- Token-based session (JWT)

B) Authorization (AuthZ)

1. Customer role

Allowed:

- Create service request
- Chat with assigned provider
- Leave reviews

Not allowed:

- Access other users' chats
- Modify provider profiles

---

2. Provider role

Allowed:

- Accept/decline service requests
- Update job status
- Chat with assigned customer
- Manage profile

Not allowed:

- Access unrelated requests