1.Executive Summary

QuickAssist is a full-stack MERN-based real-time service booking platform designed to provide instant house help services within 15 minutes.

The platform intelligently matches users with the nearest available verified service provider and displays real-time ETA using automated algorithms.

QuickAssist is designed as a scalable, secure, startup-ready solution that integrates enterprise-grade security, real-time tracking, OTP verification, and role-based access control.

2. Problem Statement

Urban households frequently require urgent assistance such as:

- Cleaning
- Plumbing
- Electric repairs
- Cooking
- Appliance repair

Current challenges:

- Delayed booking confirmations
- No instant allocation
- Uncertain arrival times
- Poor provider transparency
- Weak authentication systems

There is a strong need for a smart automated platform ensuring reliable service within 15 minutes.

3. Proposed Solution

QuickAssist provides:

- One-click instant booking
- Smart provider auto-matching
- Real-time ETA tracking
- OTP-secured authentication
- Role-based access control
- Secure JWT session management
- Email notifications
- Booking history and tracking

The system ensures fast, secure, and transparent service delivery.

4. Technology Stack

Frontend

- React.js
- Tailwind CSS
- Axios
- React Router

Backend

- Node.js
- Express.js
- MongoDB

Security & Middleware

- JWT Authentication
- bcrypt password hashing
- Cookie Parser
- Helmet
- Morgan
- Rate Limiter
- OTP Verification (Nodemailer)

5. System Architecture

User → React Frontend
↓
Express API Server
↓
Authentication Layer (JWT + OTP + Cookies)
↓
Security Middleware (Helmet + Rate Limiter + Morgan)
↓
Business Logic Layer
    • Provider Matcher
    • ETA Calculator
↓
MongoDB Database

6 Core Features

User Registration & OTP Verification

- User signs up with email & password
- OTP sent via Nodemailer
- Account activated after OTP verification
- Password stored securely using bcrypt

This prevents fake registrations and increases trust.

Secure Login System

- JWT-based authentication
- Token stored in HTTP-only cookies using Cookie Parser
- Token expiration handling
- Secure session validation

Prevents:

- Token theft
- Unauthorized access
- Session hijacking

Role-Based Access Control (RBAC)

Roles:

- User
- Provider
- Admin

Permissions:

User:

- Book services
- View booking history

Provider:

- Accept bookings
- Update service status

Admin:

- Manage users
- Monitor bookings
- View analytics

Middleware validates role before route access.

Instant Booking System

- User selects service category
- Clicks instant booking
- Smart algorithm finds nearest available provider
- Booking confirmed with unique ID

Smart Provider Matching Engine

Algorithm:

1. Filter providers by service category
2. Check availability
3. Calculate distance (Haversine formula)
4. Sort by rating + proximity
5. Assign best match

Ensures:

- Faster service
- Better quality
- Optimized allocation

ETA Calculation Engine

Formula:

ETA = Distance / Average Speed + Traffic Factor

- Distance → Haversine formula
- Average speed → 25 km/h
- Traffic factor → simulated multiplier

User sees estimated arrival time instantly.

Booking History

- Track past bookings
- Filter by status
- View details
- Cancel or reschedule

7 Security Architecture

QuickAssist implements enterprise-grade security.

Helmet

Helmet sets secure HTTP headers to protect against:

- XSS attacks
- Clickjacking
- MIME sniffing
- Content security vulnerabilities

Morgan

Morgan logs HTTP requests for:

- Monitoring traffic
- Debugging
- Detecting suspicious activity
- Production-level logging
-

Rate Limiter

Prevents:

- Brute force login attempts
- API spamming
- DoS attacks

Example Policy:

- 100 requests per 15 minutes per IP
- 5 login attempts per 10 minutes

Ensures fair usage and system stability.

Cookie Parser

- Reads JWT from secure HTTP-only cookies
- Prevents XSS token theft
- Secure session management

Nodemailer Integration

Used for:

- OTP verification
- Booking confirmation
- Password reset emails

8 Database Models

User

- name
- email
- password
- role
- verified
- bookingHistory

Provider

- name
- serviceType
- location (lat, long)
- rating
- availability

Booking

- bookingId
- userId
- providerId
- serviceType
- status
- ETA
- price
- timestamp