GenAI - Chatbot (Chatbot Evolution)

TEAM: AJAX

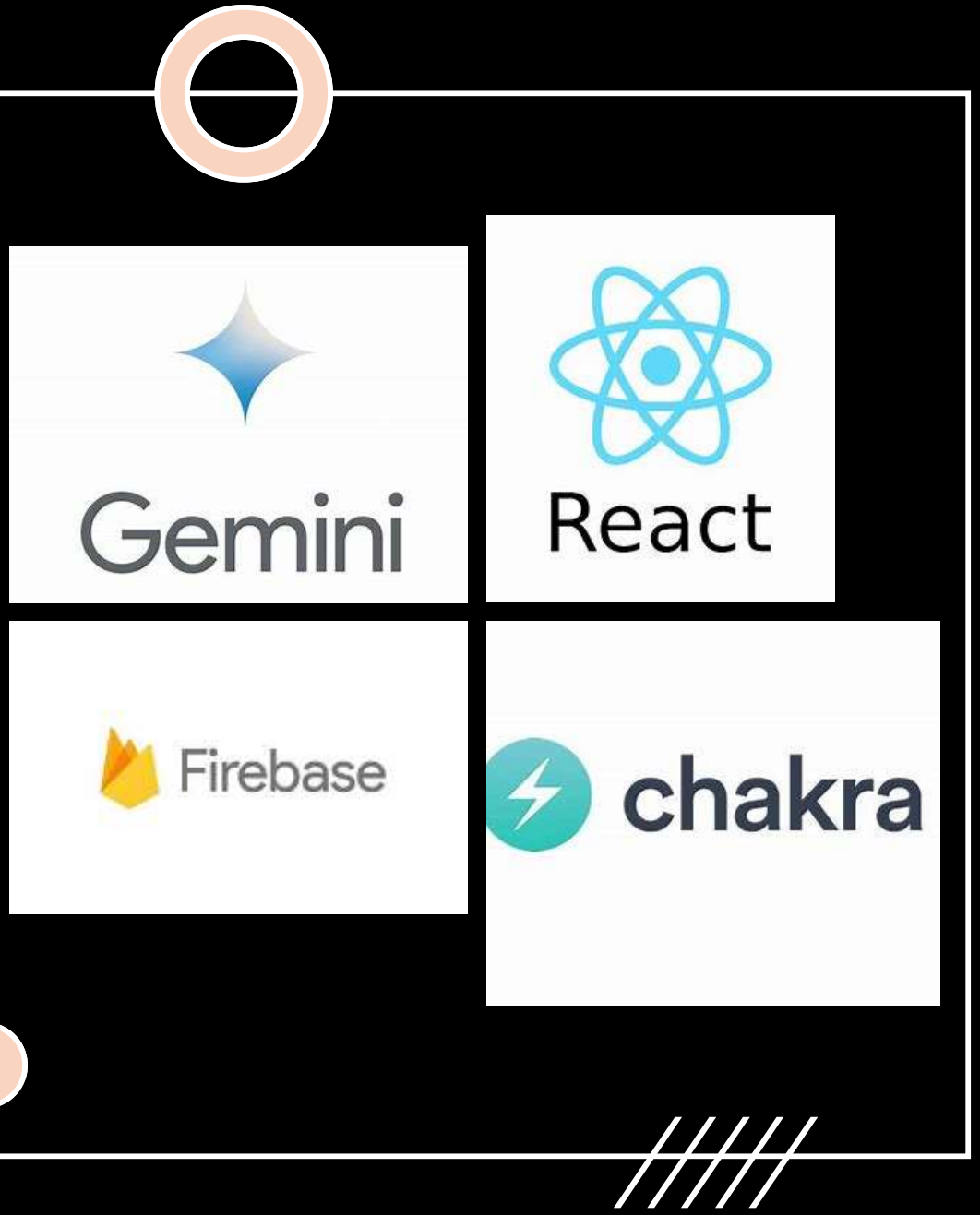
Introduction

Versatile GenAI chatbot.

Provides natural conversational interactions across multiple business domains.

Adaptable from restaurant orders to clinic appointments and hotel bookings.

Deployed on Netlify
https://chatbot-evolution.netlify.app/



Technical Excellence

- Domain Adapter Pattern.
- Smart caching and offline capabilities.
- Error boundary protection.
- Made with – React, Firebase, Google Generative AI, Chakra UI.

Key Features

Multi-Domain Support: Restaurant, Clinic, Hotel.

Smart Interactions: Google's Gemini AI, context-awareness, intelligent responses with domain data.

Multi-Language Support: Automatic detection and adaptation.

Advanced UI/UX: Responsive design, Dark/Light themes, Real-time typing indicators, Markdown support, Intuitive chat interface.

User Interface (1/1)

GenAIChatbot

Home | Restaurant | Clinic | Hotel | Sign In | Get Started

Intelligent Business Chatbot for Modern Enterprises

A cutting-edge chatbot platform showcasing innovation across multiple business domains. Our project demonstrates seamless conversational experiences for restaurant ordering, clinic bookings, and hotel reservations—all within a unified AI framework.

Try Restaurant Demo >

Try Clinic Demo >

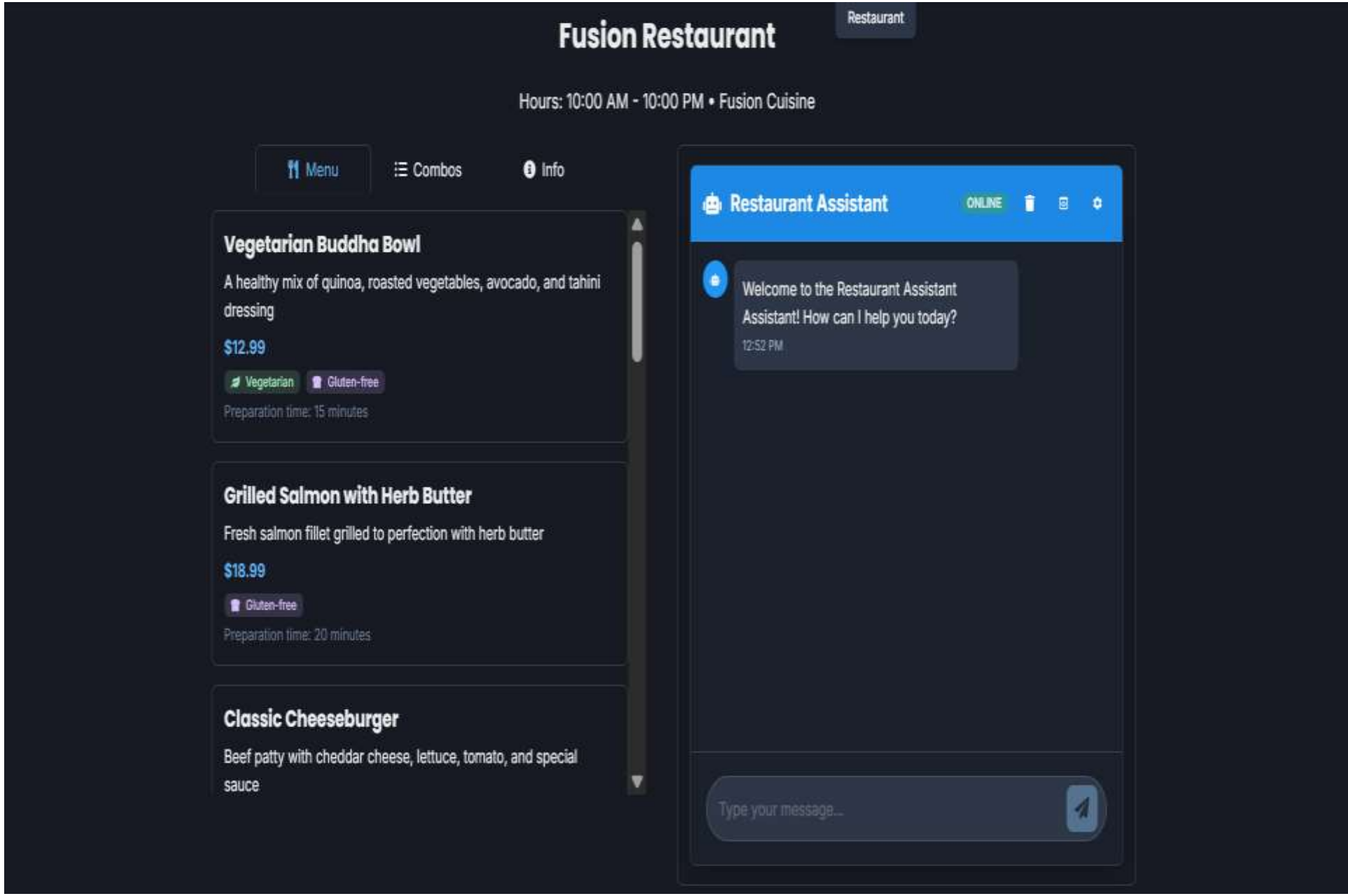
Try Hotel Demo >

SMART SOLUTIONS

MULTI-DOMAIN

INTERACTIVE

User Interface (2/2)



Versality and Domain Extensibility

- **Multi-Domain Support**
  - Currently supports Restaurant, Clinic, and Hotel domains
  - Easily extendable to new domains
  - Unified architecture across domains
- **Flexible Adaptation**
  - Dynamic tab configuration
  - Domain-specific data validation
  - Custom prompt enhancers per domain
- **Extension Mechanisms**
  - Domain registration system
  - Validation framework
  - Dynamic configuration loading
- **Implementation Examples:**
  - Restaurant: Menu & Orders
  - Clinic: Appointments
  - Hotel: Bookings
  - Future: Travel, Education, etc.

2. Domain Registration System

```
// Dynamic domain registration code
const registerDomain = (key, data) => {
  if (!availableDomains[key]) {
    availableDomains[key] = createDomainConfig(key, d
    return true;
  }
  return false;
};
```

3. Domain Loading Architecture

```
// Domain loading with validation
static async loadDomain(domainType) {
  const domainConfig = await import(`../data/${domain
  const validation = validateDomainStructure(domainCo
  // ...validation checks
  return config;
}
```

Performance and Response time Optimizations

- Smart Caching implementation
- Lazy Service Initialization
- Security Rule Optimization
- Error Handling and Recovery

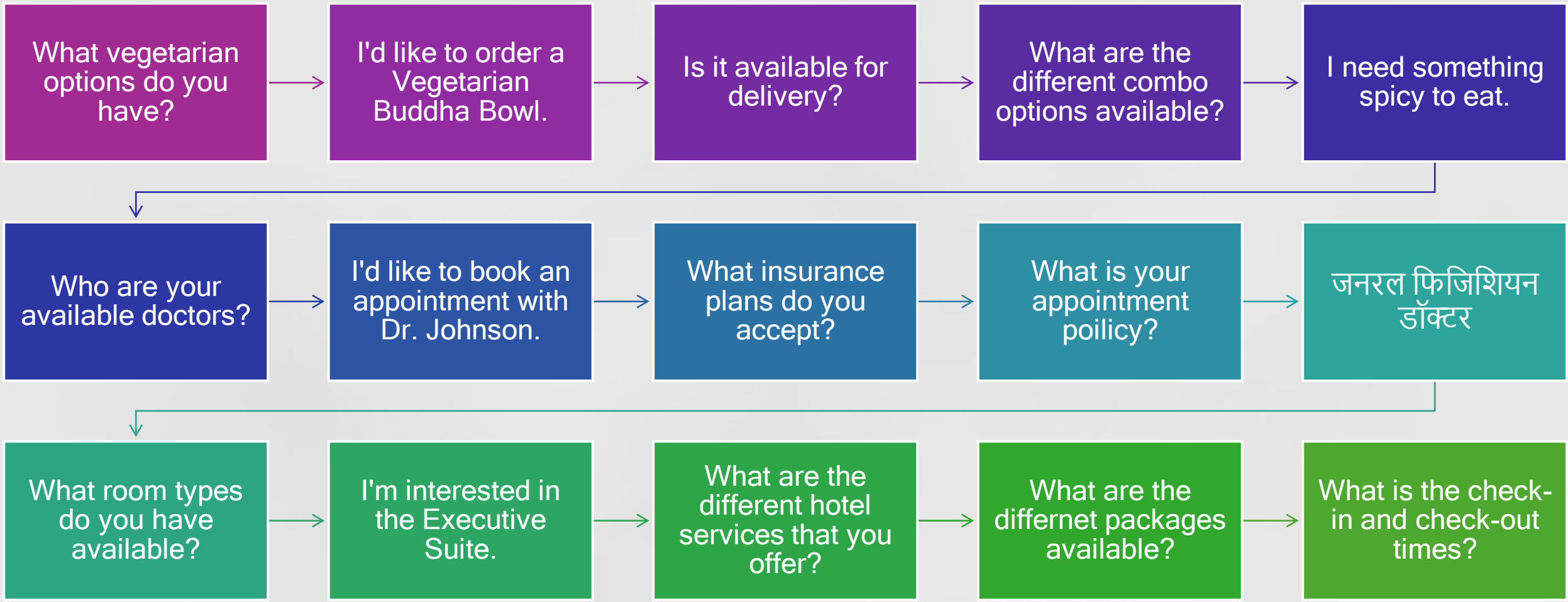
```
Example from ServiceContext.js:
useEffect(() => initializeServices(), []);
```

```
Example from optimizationService.js:
smartCache(key, getData, { ttl: 300000 })
```

Usage Examples

- **Restaurant Assistant**: Offers versatile menu and combo options.
- **Clinic Assistant**: Schedules appointments, provides doctor availability and fees.
- **Hotel Assistant**: Shows available room types and pricing.

Example of User intents



Meeting Hackathon Criteria

- Accessible and intelligent assistance.
- Integrated APIs, database and multilingual capabilities.
- Implemented 3 distinct domains, flexible structure. Solves real world problems.
- Intuitive interface, responsive design, accessibility.
- Smart caching, optimized APIs, progressive loading.
- Context-aware, dynamic responses, multi-modal inputs.


RESTAURANT
CLINIC APPOINTMENT
DOCTOR
DOCTOR
RESTAURANT
APPOINTMENT
BOOKING

Thank you