**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 29 August 2025 |
| Team ID | LTVIP2025TMID61033 |
| Project Name | Order on The Go |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

**Table-1 : Components & Technologies:**

| **Layer** | **Technology / Tools** | **Purpose in Architecture** |
| --- | --- | --- |
| **Frontend (Client App)** | **- React Native / Flutter** | **Cross-platform mobile app (iOS & Android) for customers, restaurants, and delivery partners.** |
|  | **- React.js (Web App)** | **Web interface for restaurant partners and admins.** |
|  | **- HTML5, CSS3, Tailwind** | **UI styling and responsive design.** |
| **Backend (Application Layer)** | **- Node.js (Express.js)** | **Core backend services, API handling, and business logic.** |
|  | **- GraphQL / REST APIs** | **Communication between frontend and backend.** |
| **Database Layer** | **- PostgreSQL / MySQL** | **Relational DB for users, orders, payments, and restaurants.** |
|  | **- MongoDB (NoSQL)** | **For unstructured data like menus, reviews, logs.** |
|  | **- Redis** | **Caching frequently accessed data (menu, sessions).** |
| **Payment Integration** | **- Stripe / PayPal / Razorpay** | **Secure online payments.** |
|  | **- PCI-DSS compliant gateway** | **Security and regulatory compliance.** |
| **Real-Time Services** | **- Firebase / WebSockets** | **Real-time notifications, order tracking, and chat between users & delivery partners.** |
|  | **- Google Maps API** | **Route optimization and delivery tracking.** |
| **Authentication & Security** | **- OAuth 2.0 / JWT Tokens** | **Secure user authentication.** |
|  | **- 2FA (Two-Factor Authentication)** | **Additional user security.** |
|  | **- TLS/SSL** | **Data encryption in transit.** |
| **Cloud Infrastructure** | **- AWS / Azure / GCP** | **Cloud hosting, storage, and serverless functions.** |
|  | **- Docker & Kubernetes** | **Containerization & orchestration for scalability.** |
|  | **- NGINX / Apache** | **Reverse proxy and load balancing.** |
| **DevOps & CI/CD** | **- GitHub / GitLab** | **Source code management.** |
|  | **- Jenkins / GitHub Actions** | **Automated build and deployment pipelines.** |
|  | **- Terraform / Ansible** | **Infrastructure as Code (IaC).** |
| **Monitoring & Analytics** | **- Prometheus + Grafana** | **System performance monitoring.** |
|  | **- ELK Stack (Elasticsearch, Logstash, Kibana)** | **Logging and analytics.** |
|  | **- Google Analytics / Mixpanel** | **User behavior and engagement analytics.** |
| **Notifications & Messaging** | **- Firebase Cloud Messaging (FCM)** | **Push notifications for users and partners.** |
|  | **- Twilio** | **SMS / voice call notifications.** |
|  | **- SendGrid / AWS SES** | **Email confirmations and promotions.** |

**Table-2: Application Characteristics:**

| **Characteristic** | **Requirement** |
| --- | --- |
| **Performance** | **App should load restaurant menus in <3 seconds and support 5,000+ concurrent users.** |
| **Scalability** | **Cloud-native design with containerization (Docker + Kubernetes) to handle growth in users, restaurants** |
|  | **and delivery partners.** |
| **Availability** | **99.5% uptime with load balancing and failover support.** |
| **Reliability** | **Automatic recovery from failures within 5 minutes using backup servers.** |
| **Security** | **PCI-DSS compliance, TLS/SSL for data in transit, AES-256 for data at rest, 2FA login.** |
| **Usability** | **Simple, intuitive UI; multilingual support (English + local languages).** |
| **Maintainability** | **Modular microservices-based architecture with CI/CD pipelines for easy updates.** |
| **Portability** | **Available on iOS, Android, and Web with consistent user experience.** |
| **Extensibility** | **Support for integrating new payment gateways, delivery partners, and loyalty systems in the future.** |