**Project Design Phase-II**

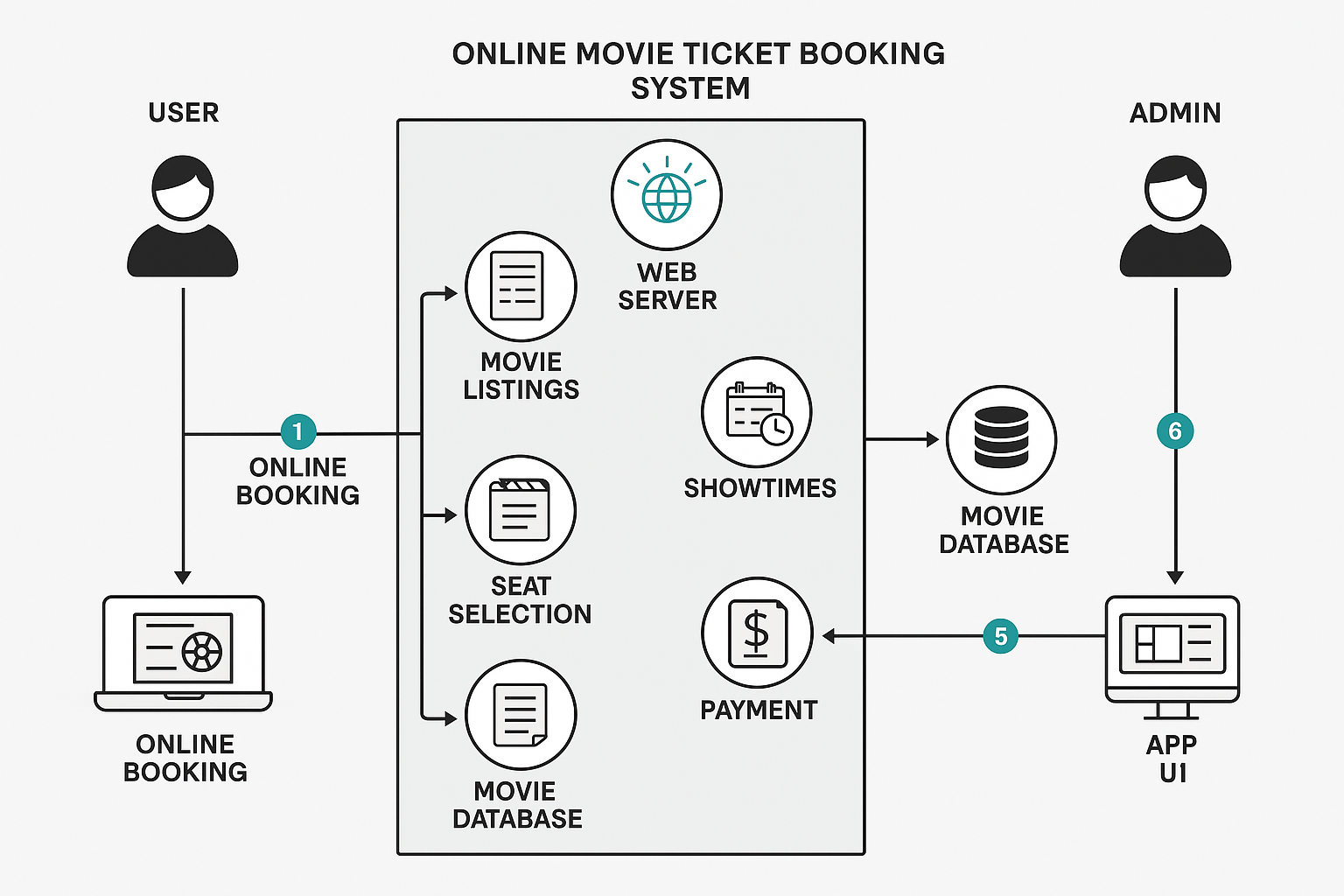
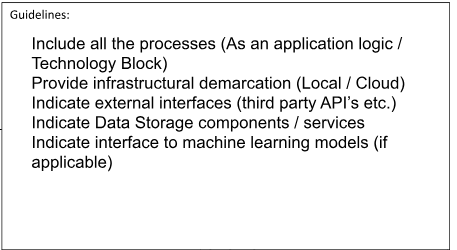
**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 25th January 2025 |
| Team ID | SWTID1744365286 |
| Project Name | Movie Ticket Booking System |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

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

**Example: Movie Ticket Booking System**

**** ****

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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
| 1 | User Interface | How user interacts with the application, e.g., Web UI, Mobile App | HTML, CSS, JavaScript / React.js / Angular.js / Flutter / Swift |
| 2 | Application Logic-1 | Logic for user authentication and profile management | Java / Python / Node.js |
| 3 | Application Logic-2 | Logic for ticket booking, seat selection, and payment processing | IBM Watson STT service, Payment Gateway APIs |
| 4 | Application Logic-3 | AI-powered chatbot for customer support | IBM Watson Assistant |
| 5 | Database | Data storage for users, bookings, movie details, etc. | MySQL, PostgreSQL, NoSQL |
| 6 | Cloud Database | Cloud-based data storage solution | IBM DB2, IBM Cloudant, Firebase Firestore |
| 7 | File Storage | Storing user-generated content, e.g., ticket receipts | IBM Block Storage, Local Filesystem, AWS S3 |
| 8 | External API-1 | Third-party API for real-time movie listings | IMDb API, TMDb API |
| 9 | External API-2 | Identity verification during booking | Aadhar API, OAuth 2.0 Authentication |
| 10 | Machine Learning Model | Personalized movie recommendations based on user behavior | Recommendation Engine (TensorFlow, PyTorch) |
| 11 | Infrastructure (Server / Cloud) | Application Deployment on Local System / Cloud | Cloud Foundry, Kubernetes, AWS, IBM Cloud |

**Table-2: Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Characteristics | Description | Technology |
| 1 | Open-Source Frameworks | Frameworks used for development | React.js, Node.js, Express.js, Django |
| 2 | Security Implementations | Security controls for user data and transactions | SHA-256 encryption, IAM Controls, OAuth, Firewalls |
| 3 | Scalable Architecture | Ensuring the system scales for high demand | Microservices, Load Balancing |
| 4 | Availability | High availability with redundancy measures | Distributed Servers, Failover Mechanisms |
| 5 | Performance | Optimization techniques for fast response times | Caching (Redis, Memcached), CDN Integration |