Project Plan: Online Movie Ticket Booking System

1. Lifecycle Model

Chosen Model:

- The Waterfall Model is appropriate for this project, considering its defined phases like requirement analysis, system design, implementation, testing, and maintenance. The project's objectives are well-defined, and the requirements are understood upfront, which fits the Waterfall approach.

Justification:

- **Clear Requirements:** The system requirements, such as online ticket booking, SMS gateway integration, and user management, are clearly outlined.
- **Sequential Nature:** Each phase has a structured sequence (requirements → design → development → testing), making the Waterfall model suitable.
- **Minimal Changes:** Since the requirements for the online booking system are straightforward, there is less likelihood of major changes during development.

2. Tools for Each Phase of the Lifecycle

Phase	Tool
Planning	Microsoft Project or Jira for project scheduling, task assignment, and tracking.
Design	UML diagrams using Lucid chart or Microsoft Visio for DFDs and ER diagrams.

Phase	Tool
Version Control	GitHub or GitLab for managing source code and tracking changes.
Development	PHP (frontend), MySQL (backend), JavaScript for dynamic client-side scripting.
Bug Tracking	Jira or Bugzilla for tracking bugs and issues during development and testing.
Testing	Selenium or JUnit for automated testing of web application components.
Deployment	Docker or Jenkins for continuous integration and deployment.

3. Deliverables: Reuse/Build Components

Reuse Components:

- **Authentication Module:** This can be reused across multiple projects, as user login and registration logic is commonly applied.
- **SMS Gateway Integration:** Third-party SMS gateways like Clickatell can be integrated into other systems with little modification.

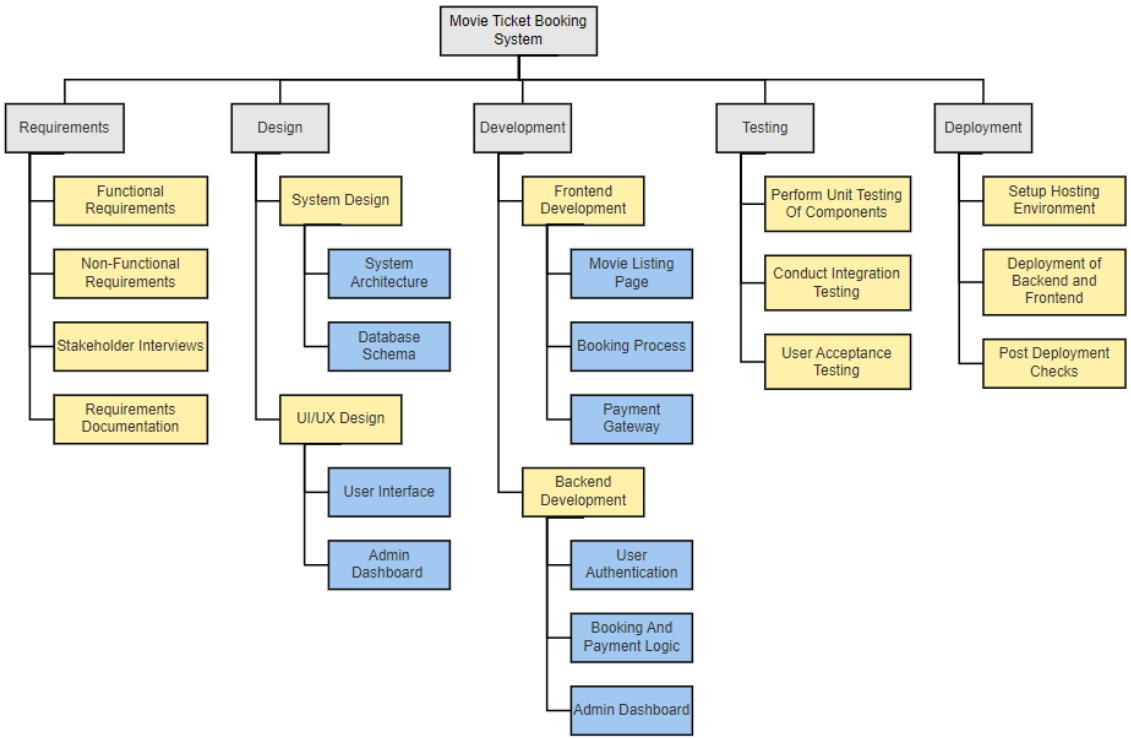
Build Components:

- **Ticket Booking Interface:** This is specific to the cinema booking system, requiring a custom design for managing showtimes and seat availability.
- **Admin Control Dashboard:** Built specifically to handle movie management and ticket tracking, requiring custom development.

Justification:

- Components like authentication and payment systems are generic and can be adapted for different systems, while ticket booking and admin control are unique to this application.

4. Work Breakdown Structure (WBS)



5. Effort Estimation and Gantt Chart

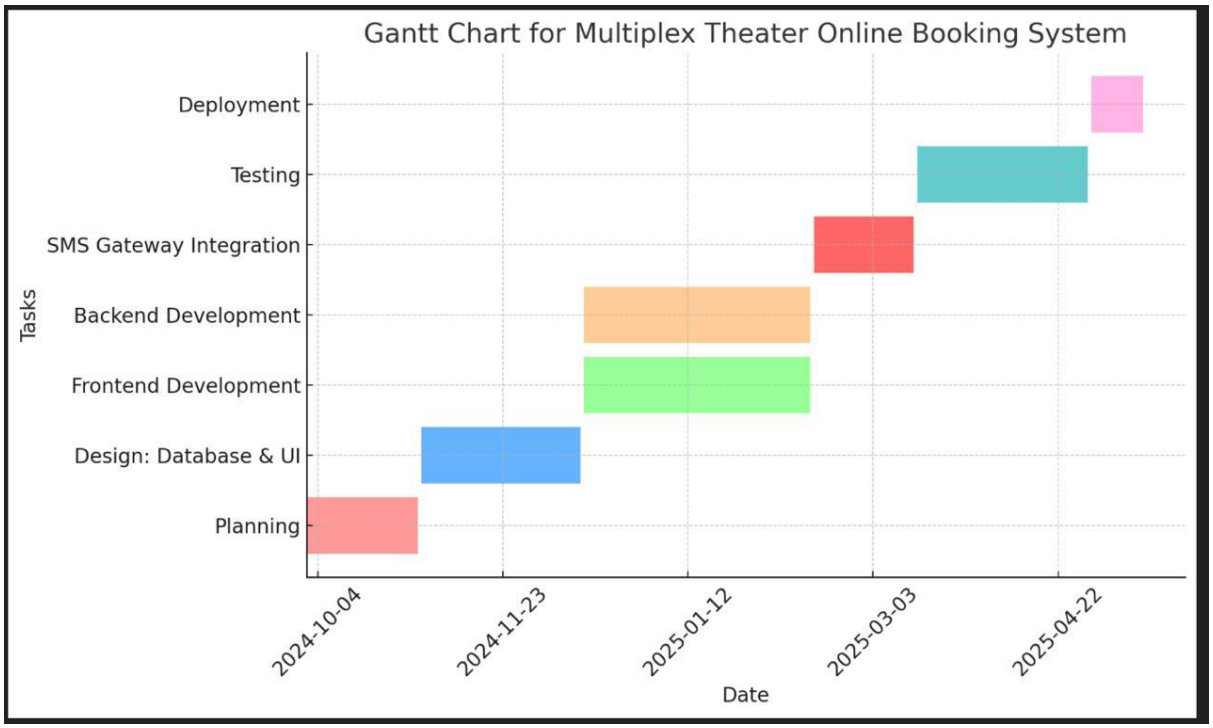
Effort Estimate:

Phase	Duration	Team Size	Total Person-Months
Planning	1 month	1 person	1 person-month
Design	1.5 months	2 people	3 person-months

Phase	Duration	Team Size	Total Person-Months
Frontend Development	2 months	2 people	4 person-months
Backend Development	2 months	2 people	4 person-months
Integration with SMS Gateway	1 month	1 person	1 person-month
Testing	1.5 months	1 person	1.5 person-months
Deployment	0.5 months	1 person	0.5 person-month

Total Effort: ~9.5 person-months

Gantt Chart:



6. Coding Details

- **Code Structure:**
  - Backend will follow an MVC (Model-View-Controller) architecture using PHP.
  - The frontend will use HTML, CSS, JavaScript, and integrate responsive design with Bootstrap.
  - Database: MySQL with tables for users, movies, showtimes, and bookings.
- **Coding Standards:** Follow the PSR-12 standard for PHP and W3C guidelines for HTML/CSS/JS.
- **Testing Strategy:**
  - Unit testing using PHP Unit for backend logic (e.g., booking flow).
  - Integration tests to validate the interaction between the frontend, backend, and the database.
  - Use Selenium for automated UI testing to ensure the seat selection and booking processes work smoothly.