Project Plan for Restaurant Management System (RMS)

1. Identify the lifecycle to be followed for the execution of your project and justify why you have chosen the model

Lifecycle Model: Agile Methodology

Justification:

- **Iterative Development**: Allows continuous feedback from restaurant managers and staff, helping to refine the system based on real-time usage and evolving requirements.
- **Flexibility**: The ability to handle changing requirements, as new modules or features can be incorporated as needed (e.g., changes in reservation management or billing).
- **User-Centric**: With regular feedback from users at the end of each sprint, the system is built to meet actual operational needs.
- **Continuous Improvement**: Aligns well with incremental feature releases like menu management, order tracking, and billing.

2. Identify the tools which you want to use throughout the lifecycle

Phase	Tools
Planning	Jira (Agile project management and sprint planning)
Design	Lucidchart or Draw.io (for designing flowcharts, DFDs, and UI mockups)
Version Control	Git + GitHub (for source code management)
Development	Visual Studio Code, Flask (for backend API development), MySQL (for database management)
Frontend	HTML, CSS, JavaScript (for client-side UI development)
Bug Tracking	Jira (for tracking issues during development cycles)
Testing	Manual testing for UI/UX and database interaction; Postman (API testing)
Deployment	Self-hosted server (or local hosting solutions like XAMPP or WAMP)

---

3. Determine all the deliverables and categorise them as reuse/build components

Reusable Components:

- **UI Templates**: Reuse standard design templates for the customer order interface and staff dashboards.
- **Database Schema**: Utilize standard schemas for managing restaurant orders, reservations, and menu items.

Justification: These components can leverage established industry solutions or follow best practices.

Build Components:

- **Custom API for Menu Management**: Build new endpoints for dynamically managing the restaurant menu.
- **Kitchen Order Coordination**: Develop a custom order management system for real-time coordination between the kitchen and customer orders.
- **Reservation System**: Build a tailored system for managing table reservations, including time slot bookings and cancellations.

Justification: These components need customization to match the specific operational needs of the restaurant.

---

4. Create a WBS for the entire functionalities in detail

Level	Task Name	Sub-tasks
1	**Restaurant Management System**	
2	Project Initialization	- Requirement gathering - Feasibility study - Architecture design
2	Backend Development	- Setup MySQL environment - API development (authentication, order placement, reservation, billing) - Database setup
2	Frontend Development	- UI/UX Design - Customer interface (order placement, reservations) - Admin interface (menu management, reports)
2	Kitchen Management	- Develop kitchen order tracking system - Real-time synchronisation with frontend
2	POS and Billing Integration	- Integrate billing with POS - Implement payment gateway

2	Testing	- Unit testing (backend) - Integration testing - Performance and load testing
2	Deployment	- Setup staging environment - Live deployment (self-hosted server) - Performance monitoring
2	Maintenance and Support	- Ongoing bug fixes - Performance enhancements

---

5. Do a rough estimate of effort required to accomplish each task in terms of personal months. Create the Gantt Chart for scheduling using any tool

Task	Effort (Person-Months)
Project Initialization	1
Backend Development	3
Frontend Development	2.5
Kitchen Management	2
POS and Billing Integration	1.5
Testing	1.5
Deployment	1
Maintenance and Support	0.5
Total Estimated Effort	**12 Person-Months**

Here’s a rough Gantt chart based on the tasks:

Phase/Task	Month 1	Month 2	Month 3	Month 4	Month 5
Project Initialization	X				
Backend Development	X	X	X		
Frontend Development		X	X		
Kitchen Management		X	X		

POS and Billing Integration			X	X	
Testing			X	X	
Deployment				X	
Maintenance and Support				X	X

---

6. Coding Details

- **Languages**: JavaScript (for frontend), HTML/CSS (for styling), Flask (for backend), MySQL (for database).
- **API Design**: RESTful APIs for order placement, menu management, and billing.
- **Modules**:
    - **Authentication**: Handles login/logout for staff and admin.
    - **Menu Management**: API for adding, updating, and removing menu items.
    - **Order Management**: Real-time order tracking from customer placement to kitchen completion.
    - **Reservation Management**: Table booking with date and time slots.
    - **Billing**: Integration with POS systems and payment gateways.