**Project Management System**

1. The Project Management System is designed to improve project efficiency by organizing tasks and tracking progress. It allows project managers to assign tasks to employees, maintain schedules, and provide real-time progress updates. The system also includes features like user access controls and a feedback loop, enabling a collaborative and transparent project management process.

2. **Project Assignment:** When a new project is initiated, the project manager assigns specific tasks to employees based on their roles and skills.

3. **Admin Module:** The admin module is central to this system. It allows the admin to assign tasks and manage the project schedule, making it easier to coordinate everything in one place.

4. **User Access:** To ensure project transparency, we created secure user accounts. Employees and managers can log in to view the project’s progress, which is continuously updated by the manager.

5. **Feedback Team:** A dedicated feedback team is involved at each development stage to gather feedback. This feedback process helps improve the project as it progresses.

In summary, the system helps keep projects on track by allowing the admin to manage tasks and schedules, enabling users to monitor project progress, and incorporating feedback for continuous improvement.

**1. Can you describe the purpose of your Project Management System?**

• Answer: The Project Management System is designed to improve project efficiency by organizing tasks and tracking progress. It allows project managers to assign tasks to employees, maintain schedules, and provide real-time progress updates. The system also includes features like user access controls and a feedback loop, enabling a collaborative and transparent project management process.

**2. What technologies did you use in this project, and why?**

• Answer: We used JavaScript for a dynamic and responsive user interface, and Spring Boot as the backend framework for handling business logic and server-side processes. Hibernate was used as an ORM tool to simplify database interactions with MySQL, which served as our primary data storage solution. Lastly, we deployed the application on AWS Cloud for scalability and security.

**3. How does the system handle task assignment and management?**

• Answer: In our system, tasks are assigned through the Admin Module, which centralizes task assignment and project scheduling. The project manager or admin can allocate tasks to employees based on their roles and skills, making it easier to manage the entire project from one place.

**4. Can you explain the role of the feedback team in this system?**

• Answer: The feedback team plays a critical role in quality control. At each development stage, the team collects feedback to identify potential improvements or issues. This helps ensure that the project meets quality standards and adapts to any required changes as development progresses.

**5. How is data managed and stored in your Project Management System?**

• Answer: We used Hibernate with MySQL as our database. Hibernate automatically handles the mapping between our Java objects and MySQL tables, making data storage and retrieval efficient. This setup allows us to maintain project, user, and task data reliably.

**6. What role does AWS Cloud play in your project?**

• Answer: We deployed the application on AWS Cloud to ensure scalability and security. AWS provides the infrastructure to handle increasing traffic and secure data storage, which is essential as more users join the system and project data grows.

**7. How does the system ensure security and transparency for users?**

• Answer: Each user has secure account access, allowing them to log in to view project progress. This transparency allows employees and managers to stay informed about the current status of tasks and projects, promoting accountability and collaboration across teams.

**8. Can you give an example of how Hibernate improves efficiency in your system?**

• Answer: Hibernate reduces the need to write complex SQL queries by automatically mapping Java classes to database tables. This ORM tool simplifies CRUD operations and data retrieval, allowing us to focus on business logic rather than database management. For example, saving project details or retrieving assigned tasks is straightforward with Hibernate’s repository interfaces.

**9. How does the Admin Module enhance project coordination?**

• Answer: The Admin Module serves as the command center for project managers. It enables them to assign tasks, monitor deadlines, and manage schedules. With everything in one place, it’s easy to keep projects organized and ensure that all tasks are on track.

**10. What challenges did you face while developing this system, and how did you overcome them?**

• Answer: One challenge was managing real-time updates for project progress. We solved this by using efficient backend logic in Spring Boot and leveraging JavaScript for dynamic front-end updates. This way, users always see up-to-date information without frequent page reloads.

**Tourism and Hospitality Management System**

The Hospitality Management System is a web-based application that helps hotels to run smoothly by managing room bookings, guest check-in/check-out, billing, and customer service.

Basically there are 2 modules in the project one is the Admin and the User

Admin: Admin is responsible for adding a new Tourist site, new tour packages into the website and verifying and validating the user registration. Tracking all the booking made by the user like rentals, also adding all the needed hospitality like rooms and hotels for stay at a particular tourist attraction and enabling bookings.

User: User is responsible for registration and creating account in the website. Users can avail information about all the tourist sites. A login page to login and see the booking made related to a tour package or rentals or stays. User will be getting an email conformation after successful registration and booking.

I was responsible for implementing the Admin module developed all the tasks listed above like validating users, dynamically adding about tourist attrcations and stays and tracking all the user booking.

Implemented Security features like Authentication and Authorization, Session time outs and Role Based Access.

1. **Q: What are the main technologies used in this project, and why were they chosen?**
   * **A:** We chose **Spring Boot** for its efficiency in building REST APIs and ease of integrating with Spring Data JPA for database management. **HTML, CSS, and Bootstrap** were used for the frontend to create a responsive, user-friendly interface. For the database, we used an **SQL database** as it provides robust data handling and supports complex queries, which are essential for operations like reservation management and billing.
2. **Q: How does the Reservation Management module handle real-time room availability?**
   * **A:** The **Reservation Management module** maintains a table of room statuses and updates them in real-time with each new booking or cancellation. Whenever a booking request is made, the system checks for available rooms by querying the database. This allows us to confirm availability immediately and prevents double-booking.
3. **Q: How did you implement secure access control in this project?**
   * **A:** We implemented **role-based access control** by defining user roles such as front desk, housekeeping, and admin. Each role has specific permissions within the system. For example, front desk staff have access to check-in/check-out features, while admins manage room rates and system settings. We used **JWT (JSON Web Tokens)** to authenticate and authorize users, ensuring secure access to resources.
4. **Q: How does the billing system work in your project?**
   * **A:** The **Billing and Invoicing module** automatically calculates charges for rooms, services, and amenities. At checkout, the system generates an invoice by summing up all charges incurred during the guest's stay. We used SQL queries to fetch room rates and additional charges, then formatted the data into a detailed bill, which can be printed or emailed to the guest.

**Q: What were the biggest challenges you faced when working with Spring Boot, and how did you overcome them?**

* + **A:** One challenge was managing **database transactions** to ensure data consistency, especially with reservations and billing. We used Spring Boot’s **@Transactional annotation** to handle transactions, ensuring that complex operations either fully complete or rollback in case of an error. Debugging dependency issues with Spring Boot libraries was another challenge, which we resolved by carefully managing dependencies in our pom.xml file.

**Design and Architecture Questions and Answers**

1. **Q: Can you explain the overall architecture of your system?**
   * **A:** The architecture consists of three main layers: the **frontend**, built with HTML, CSS, and Bootstrap for user interaction; the **middleware** with Spring Boot, which handles the business logic and processes requests; and the **SQL database**, where all application data is stored. The frontend interacts with the middleware via REST APIs, while the middleware interacts with the database through Spring Data JPA for ORM.
2. **Q: How did you ensure data consistency across different modules?**
   * **A:** Data consistency was maintained through careful transaction management in **Spring Boot**. For critical operations, like booking or billing, we used Spring’s @Transactional feature to control the transaction boundaries. Additionally, we implemented checks to prevent duplicate bookings and validate data inputs, reducing the risk of inconsistent data.
3. **Q: What would you improve in this system if given more time?**
   * **A:** I would add more **data analytics capabilities** to help management make data-driven decisions. For example, providing insights into guest preferences, occupancy trends, and revenue generation could help optimize pricing and enhance customer service. I would also explore adding a **cloud-based database** to support potential scalability for larger hotel chains.

**Teamwork and Project Management Questions and Answers**

1. **Q: What was your role in this project, and what were your main contributions?**
   * **A:** I was responsible for implementing the **Reservation Management and Billing modules**, as well as integrating the database with Spring Data JPA. My main contributions included writing the logic for handling bookings, managing data transactions, and creating the automated billing system. I also collaborated closely with the team to design the frontend interface and ensure smooth communication between modules.
2. **Q: How did you manage communication and coordination within the team?**
   * **A:** We used **GitHub** for version control and collaborated on feature branches to prevent conflicts. We held **weekly meetings** to discuss progress, resolve any issues, and coordinate our tasks. We also communicated via a messaging app, which helped us stay aligned and share updates as needed.
3. **Q: What challenges did you face as a team, and how did you address them?**
   * **A:** One challenge was aligning our code styles and ensuring consistency across modules. We resolved this by setting coding standards early on and using code reviews to maintain quality. Another challenge was managing deadlines with multiple modules being developed simultaneously. We prioritized key features and used a shared **project tracker** to stay on schedule.