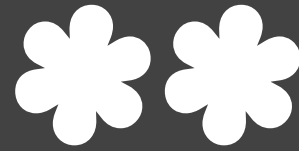


La

Brasserie



Presenters: Rose Gekonge, Hibrahim Onyango, Fautomata Sarr

Presentation Date 29/08/2024



The Team



Rose Gekonge



Hibrahim Onyango



Fatoumata Sarr

Project Description:

La Brasserie is a five-star hotel in Dakar that epitomizes luxury and sophistication. The hotel offers exquisitely designed rooms and suites with modern amenities and stunning views. Guests can indulge in gourmet dining at world-class restaurants, relax at the luxurious spa, stay active in the fitness center, and enjoy both indoor and outdoor pools. The hotel provides exceptional personalized services, a dedicated concierge, and state-of-the-art conference facilities for business travelers. Located in the vibrant heart of the city, the hotel is perfect for exploring cultural landmarks and premier shopping districts. Committed to sustainability, the hotel implements eco-friendly practices to protect the environment. **La Brasserie** also features elegant event spaces, making it ideal for weddings and special occasions, ensuring every stay is memorable.



Project Description:

The primary goal of this project was to develop a web application tailored for managing hotel operations. This application covers various functionalities such as room bookings, customer management, and other essential hotel services. By using Flask for the backend and SQLAlchemy for database management, I aimed to create a reliable, user-friendly platform that streamlines hotel operations.

My Role:

As a team, we were responsible for the entire development process, including planning, design, backend development, frontend integration, database management, testing, and deployment.

Architecture Diagram :

The architecture of the application consists of multiple components working together seamlessly. The diagram illustrates the following:

- **Frontend:** Built using HTML, CSS, and JavaScript, the frontend is designed to be responsive and user-friendly, allowing customers to interact with the hotel services easily.
- **Backend:** The backend is powered by Flask, which handles all server-side logic, including routing, session management, and API integration.
- **Database:** SQLAlchemy serves as the ORM, managing interactions with a relational database using SQLite. It handles the creation and management of tables related to rooms, bookings, customers, and more.
- **APIs:** Integration with third-party APIs, such as payment gateways or map services, enhances the functionality of the application.
- **Hosting:** The application is deployed on Github, providing a scalable and reliable environment for hosting the web application.

The Technology :

❖ Backend Technologies:

- Flask: Flask is the backbone of the server-side operations, handling HTTP requests, managing application routes, and processing business logic. Its simplicity and flexibility made it an ideal choice for this project.
- SQLAlchemy: I used SQLAlchemy to interact with the database, allowing me to perform complex queries and manage data efficiently. SQLAlchemy's ORM capabilities ensured that the database operations were seamless and aligned with Python code.

❖ Frontend Technologies:

- HTML/CSS/JavaScript: These technologies were used to build the user interface, ensuring that the application is responsive and intuitive. CSS was particularly important for styling, while JavaScript enabled dynamic content and interactions on the client side.

❖ Deployment tool

- The application is deployed on Github, providing a streamlined deployment process with amazing features.

Development Process

Project Phases:

- **Planning:** We began by defining the project's scope and requirements. This included identifying key features such as room booking, customer management, and payment processing. I also outlined the database schema and the overall architecture of the application.
- **Design:** The design phase involved creating wireframes and mockups for the user interface. We focused on user experience, ensuring that the application is easy to navigate and visually appealing. We also designed the database schema, defining the relationships between different entities such as rooms, bookings, and customers.
- **Development:** During development, We implemented the backend logic using Flask and SQLAlchemy. We created models for each database table, routes for handling user requests, and templates for rendering dynamic content. The frontend was developed concurrently, integrating with the backend to provide a seamless user experience.
- **Testing:** We conducted thorough testing throughout the development process. Unit tests were written to ensure the accuracy of individual components, while integration tests verified that the system worked as a whole. I also performed user acceptance testing to gather feedback from potential users.

Tools Used:

- **VS Code & Github:** it was used for version control, managing code changes and collaborating effectively if working with a team. GitHub was used as the remote repository, allowing for easy collaboration and deployment.
- **Kanban:** It was used for project management and planning.

The Success

- ★ Successful Integration of Flask and SQLAlchemy: We effectively integrated Flask with SQLAlchemy, allowing for smooth database interactions and efficient data management.
 - ★ Implementation of a Robust Booking System: The booking system was a critical component of the application. We successfully developed a system that allows users to check room availability, make reservations, and manage bookings.
 - ★ Positive Feedback from User Testing: The application received positive feedback from users during testing. Users appreciated the simplicity and responsiveness of the interface, as well as the ease of navigating through different functionalities.
 - ★ Efficient Database Management: Using SQLAlchemy's ORM, We were able to manage the database effectively, ensuring data consistency and integrity across the application.
 - ★ Successful Deployment: The application was successfully deployed on Github, with all features functioning as expected in the live environment.
-

The problem

- ★ Communication barrier among the team members since Sarr is from senegal and the rest are from Kenya.
- ★ Time Management
- ★ Difference in opinions on various methods and how to handle them to solve respective tasks.
- ★ Time difference over the different countries.
- ★ Ensuring Data Consistency: Maintaining data consistency in the database, especially when dealing with concurrent bookings and updates, was challenging. We implemented transaction management and locking mechanisms to prevent data corruption.
- ★ Handling Edge Cases in Booking Logic: Ensuring that the booking logic handled all possible scenarios, such as overlapping reservations or cancellations, required careful consideration. We addressed these challenges by implementing comprehensive validation checks and unit tests.

How I Overcame Them:

- We approached these challenges by leveraging online resources, collaborating with peers, and performing extensive testing. Each challenge provided valuable learning opportunities, contributing to the overall success of the project

Areas for Improvement

- **Code Optimization:**

There is potential to refactor and optimize parts of the codebase. For example, some database queries could be optimized for better performance, especially as the application scales.

- **User Interface:**

The UI could be enhanced with more sophisticated design elements and animations. Additionally, adding more features to improve user experience, such as a recommendation system or user reviews, would be beneficial.

- **Scalability:**

The current implementation is designed for small to medium-sized hotels. To accommodate larger operations, We plan to enhance the application's scalability, including optimizing database performance and improving the handling of concurrent bookings.

Lessons Learned

- **Technical Lessons:**

Through this project, we gained a deeper understanding of Flask and SQLAlchemy, particularly in the context of building real-world applications. We also learned about best practices in web development, such as writing modular code, implementing secure authentication, and managing databases effectively.

- **Project Management:**

Managing this project from start to finish taught me the value of effective planning and time management. By breaking down tasks into manageable chunks and setting clear milestones, I was able to stay on track and deliver a successful application.

Next Step

Future Plans:

- Looking ahead, we plan to add new features, such as payment integration for a more seamless booking experience and customer reviews to enhance the application's value. We also aim to implement additional security measures, such as two-factor authentication and encryption of sensitive data, to protect user information.

Continuous Improvement:

- We are committed to continuously improving the application by regularly updating it based on user feedback. This includes adding new features, fixing any issues that arise, and optimizing the application's performance.

WORK PLAN

by kanban Board

[SCHEDULE OF THE WORK]



To-do	Do today	In progress 1 / 3	Done
<div><div>Establish a strong foundation of our Knowledge on all technologies we are meant to use for the project</div><div><div>HO</div><div>RG</div><div>FA?</div></div><div>Due: Saturday 5:00 PM (Done)</div></div> <div><div>DOM of the project & Pseudocodes</div><div><div>HO</div><div>RG</div><div>FA?</div></div><div>Due: 12 August 9:00 PM (Done)</div></div> <div><div>Front End Establishment</div><div><div>FA?</div></div><div>This is all about the viewable part of the project and its impression.</div><div>Due: 17 August 10:00 PM (Done)</div></div> <div><div>Backend Establishment and Databases</div><div><div>HO</div><div>RG</div></div><div>This is everything about the functionality and execution of the intended response and workflow. It is mostly about adding flesh to the skeleton.</div><div>Due: 17 August 12:00 PM (Done)</div><div><div>Framework & Server</div><div>Due: Aug 14 at 11:00 PM</div></div><div><div>Database</div><div>Due: Aug 17 at 11:00 PM</div></div></div> <div><div>Debugging & Launching</div><div><div>HO</div><div>RG</div><div>FA?</div></div><div>Due: 19 August 5:00 PM (Done)</div></div> <div><div>Testing and Accessing</div><div><div>FA?</div></div><div>This will be done by random friends or family</div><div>Due: 20 August 5:00 PM (Done)</div></div>	<div><div>Web Portfolio Pitch</div><div><div>HO</div><div>RG</div><div>FA?</div></div><div>Let's Plan!!!</div></div> <div><div>Establish a strong foundation of our Knowledge on all technologies we are meant to use for the project</div><div><div>HO</div><div>RG</div><div>FA?</div></div><div>Due: Saturday 5:00 PM (Done)</div></div>	<div><div>Web portfolio Pitch</div><div><div>HO</div><div>RG</div><div>FA?</div></div><div>As mentioned in the beginning of this project, you have some questions to ask yourself.</div><div><ul style="list-style-type: none">What do I want to build?How do I want to build it?</div></div>	

CONCLUSION

This project has been a rewarding experience, allowing me to apply my skills in Flask, SQLAlchemy, and web development to build a functional and user-friendly hotel management application. The successes achieved, challenges faced, and lessons learned have all contributed to my growth as a developer.

Thank you for your attention. We are happy to answer any questions and discuss any aspects of the project in more detail.