VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Project Report

on

Online Flight Ticket Booking System Using Microsoft Azure

to be submitted in partial fulfillment of the requirements for the course on

SWE4002 - Cloud Computing
Slot - (B2)

Review - 3

by

Chinthaparthi Archana	**21MIS0211**
Renuka A	**21MIS0006**
Hithaishini K	**21MIS0041**

Under the guidance of

Prof. Harshita Patel

SCORE - School of Computer Science Engineering and InformationSystem

Winter Semester 2023-2024

TITLE OF THE PROJECT:

Online Flight Ticket Booking system using Microsoft Azure

ABSTRACT

This project presents an online flight ticket booking system using Microsoft Azure. The primary purpose of this system is to simplify the process of booking flight tickets for users and streamline the management of booking information for airline staff and administrators.

The methodology involved building a web-based application where users can select their departure and arrival dates, choose from available flights, and book tickets accordingly. The airline staff has access to view booked tickets, monitor the schedule, and check available seats. Administrators are provided with control to add or remove users, including staff and other admins.

By leveraging Microsoft Azure, the system benefits from scalable infrastructure, enhanced security, and high reliability, ensuring a smooth experience for users and staff. Our key findings show that using Azure as a cloud platform for deployment provides a robust solution, allowing for easy management and secure data storage. This project demonstrates the significance of cloud-based solutions in modern web applications, especially for industries like airline booking where reliability and security are paramount.

INTRODUCTION

The airline industry has experienced rapid growth in recent years, with more people traveling by air than ever before. This surge in demand has created a need for efficient and user-friendly systems for booking flight tickets online. Traditional booking methods, such as calling an airline or visiting a travel agency, are often time-consuming and inconvenient.

In response to this problem, we developed an online flight ticket booking system that simplifies the booking process for users while providing airline staff and administrators with tools to manage flight schedules and ticket sales. The system is designed to be user-friendly, allowing customers to book tickets with a few clicks, choose their preferred flights, and receive confirmation instantly.

The context for our work lies in the increasing reliance on digital platforms for various services. By leveraging cloud computing, specifically Microsoft Azure, we aimed to create a robust and scalable system that could handle high traffic while maintaining security and reliability. Azure's flexibility and resources made it an ideal platform for deploying our online flight ticket booking system, ensuring a seamless experience for users and efficient management capabilities for airline staff and administrators. This project contributes to the broader trend of digital transformation in the airline industry and showcases the potential of cloud-based solutions in enhancing customer experiences and operational efficiency.

LITERATURE SURVEY

S.NO	TITLE	YEAR OF PUBLICATION	METHODOLOGY USED	DESCRIPTION
1.	Architecture Model for Flight Booking System Based on Web Services	2017	SOAP ,Apache web service for interface design	To propose an architecture model for flight booking system depend on Web services and it can serve as a guideline for online flight booking application for end users to book flight tickets. It can be concluded that this model generally has a moderate strength to achieve this system.
2.	Uses of a Cloud-Based Flight Management System to Enhance Airspace Efficiency	2022	Cloud (FMS) digital twin of FMS	It has access to the state of the aircraft on which the FMS is running, it contains a wealth of information not normally available to ground-based aviation systems. In addition, the CFMS can access large amounts of compute power that is unavailable to the flight-deck based FMS.
3.	Implementing a Cloud-Based Flight Management System	2017	Four dimentional trajectory	This enables Trajectory-Based Operations (TBO) by sharing the flight plan (planned route) as well as aircraft state with a secure cloud environment, enabling accurate trajectory prediction. Additional computations such as wake vortex estimation and ground noise footprints are feasible with Cloud FMS but infeasible with a traditional FMS
4.	An enhanced cloud based model for flight data recorder(FDR)	2018	Cockpit Voice Recorder (CVR), Quick Access Recorder (QAR)	It cross checks the flight data and provides a clue to the condition of a flight that is air bound. All flights with an abnormal condition are flagged red and this allows flight control tower administrators to alert the aircraft concerned as well as alert some emergency units for a standby given the location of the flight.

5.	Demonstrating a Cloud-Enabled Flight Management System	2023	FMS via a high-speed, low-latency air-ground network	It provides architecture and concept can be used for use cases to improve the efficiency and flow of the National Airspace System (NAS), including new entrants. This concept can also be applied to Advanced Air Mobility concepts for similar purposes. In addition, it could be a feasible architecture to implement the concept of Digital Flight currently
6.	Cloud-based flight delay prediction using logistic regression	2017	Predicting flight delays via cloud-based logistic regression	Airlines play a crucial role in transporting people and goods on time, and delays can negatively impact work and business. Forecasting delays is essential for commercial airlines. This paper uses logistic regression and Microsoft Azure Learning Studio to predict aircraft departure times. Weather data is combined with airport data to derive more accurate predictions. The method achieved about 80 percent accuracy in predicting flight delays based on past training.
7.	A cryptographic cloud-based approach for the mitigation of the airline cargo cancellation problem	2020	Crypto-cloud mitigation of airline cargo cancellation	Airline Cargo companies don't charge for last-minute shipments, allowing customers to book the same shipment on multiple companies. However, uncertainty in cancellations makes it difficult to fine-tune the optimal overbooking level, causing losses. This work proposes using cryptographic techniques to improve service chains, extracting relevant information while maintaining customer data privacy.
8.	Improved wild horse optimization with levy flight algorithm for effective task scheduling in	2023	Efficient cloud task scheduling through enhanced wild horse optimization with Levy flight algorithm.	The Improved Wild Horse Optimization (IWHO) algorithm aims to enhance task scheduling efficiency in cloud computing by combining wild horse optimization with Levy flight theory. This approach minimizes Makespan and maximizes

	cloud computing			resource utilization, outperforming other methods in various scenarios.
9.	Automating the Indian transportation system through intelligent searching and retrieving with Amazon Elastic Compute Cloud	2021	Automating Indian transport via intelligent search & retrieve using Amazon EC2 cloud technology, in one line.	This article discusses issues with the Indian transportation system and proposes a method for web applications to search for data and generate shortest and lowest cost travel routes. The method uses semantically correlated gathering terms separate from search results, reducing customer dependency on vehicle services. The proposed intelligent algorithm is analyzed for effective data searching, demonstrating suitable performance against peer methods.
10.	A formal model for the interoperability of service clouds	2015	Formal interoperability model for service cloud methodology in one line: unified, efficient service integration.	This paper investigates the formal high-level specifications of large-scale service-oriented computing, focusing on the "service cloud" concept. It extends Abstract State Services (AS2s) by using high-level action schemes called "plots" to specify service operations. The paper develops a model for service mediators, ensuring service slots are filled by actual services. Matching conditions for mediators and services require a generalized projection and a semantically adequate service ontology.

PROJECT DEFINITION

The "Online Flight Ticket Booking System Using Microsoft Azure" is designed to create a reliable, scalable, and user-friendly platform for booking flight tickets. Using Microsoft Azure as the underlying cloud platform, the system will enable real-time ticket availability, secure transactions, and a seamless user experience accessible across various devices.

PROBLEM STATEMENT

Many existing flight ticket booking systems face issues with reliability, security, and scalability, especially during high-demand periods. Traditional server-based booking platforms often encounter downtime, slow response times, or security vulnerabilities, leading to a suboptimal user experience and potential security risks. Furthermore, these systems may lack integration with other services, complicating users' efforts to find essential information such as flight schedules, pricing, or additional

services like car rentals and accommodations. This fragmented experience can frustrate users, lead to lost business, and reduce overall customer satisfaction.

The impact of these issues is considerable. System downtime or sluggish performance can drive customers to competitors, resulting in revenue loss for airlines and booking platforms. Security vulnerabilities increase the risk of sensitive customer data exposure, which can lead to identity theft or financial fraud. A fragmented booking experience, marked by incomplete or inaccurate information, further erodes user trust and confidence in the platform. Moreover, conventional booking systems may struggle to scale during peak seasons, causing system overload and undermining customer satisfaction.

The proposed project aims to address these challenges by building an online flight ticket booking system that leverages the cloud-based scalability and security features of Microsoft Azure. This solution intends to reduce downtime, enhance security, and deliver a reliable platform for booking flights and related services, ultimately boosting customer satisfaction and business success.

OBJECTIVES

The "Online Flight Ticket Booking System" aims to create an efficient, reliable, and secure platform for booking flights. By leveraging Microsoft Azure for deployment, this system can ensure scalability, security, and ease of maintenance, essential for cloud computing projects. The primary objectives are:

Scalability: Utilize Azure's cloud infrastructure to ensure the platform can handle fluctuations in user demand without performance degradation. The system should be able to scale dynamically during peak booking times.

Security: Implement strong security features through Azure's built-in security services, ensuring user data protection and secure transactions. This includes encrypted data storage, secure authentication, and compliance with industry security standards.

Availability: Ensure high system uptime by using Azure's reliability features, such as automatic backups, disaster recovery, and redundant resources to minimize downtime.

User Experience: Provide a straightforward and user-friendly interface for booking flights, allowing users to select departure and arrival dates, and facilitating a smooth booking process.

Staff and Admin Functionality: Allow staff members to view booked tickets, available seats, and flight details, while enabling administrators to manage users by adding or deleting accounts as needed. This facilitates efficient operations and user management on the cloud platform.

FUNCTIONAL REQUIREMENTS

The functional requirements focus on delivering the core features for a flight booking system while leveraging the capabilities of Microsoft Azure for cloud-based deployment and management. The key requirements are:

Flight Booking for Users: Allow users to book flights by selecting departure and arrival dates. The system should display available flights and allow users to complete bookings with a seamless checkout process.

Staff Viewing and Management: Provide staff members with a user-friendly interface to view booked tickets, monitor remaining available seats, and access other relevant flight information. This feature is essential for operational efficiency.

Admin User Management: Enable administrators to add or delete user accounts, including staff and other administrators. This functionality provides a secure way to manage access and maintain control over the system's user base.

Azure-Based Deployment and Maintenance: Deploy the system on Microsoft Azure to take advantage of cloud-based scalability, security, and reliability features. Azure's flexibility allows for easy maintenance and updates, ensuring the system stays current with user needs and technology trends.

WEBSITE HOSTING

The website hosting for the "Online Flight Booking System" project involves deploying the web application on Microsoft Azure, providing a robust and scalable environment for hosting. Azure offers a wide range of services and features that cater to website hosting, ensuring that the system is reliable, secure, and capable of handling varying levels of user traffic.

By hosting the website on Azure, the project benefits from the following key aspects:

Scalability: Azure's cloud infrastructure is designed to scale resources based on demand. This allows the website to handle increased traffic during peak booking times without compromising performance. With Azure App Service, the website can automatically scale up or down depending on user activity.

High Availability: Azure ensures high availability through redundant infrastructure and load balancing. This minimizes downtime and ensures a smooth experience for users booking flights, even during maintenance or updates.

Security: Azure provides robust security features, including Azure Active Directory for authentication and access control, Azure SQL Database for secure data storage, and built-in DDoS protection. These features help protect sensitive booking information and user data from unauthorized access or cyber threats.

Ease of Deployment: Using tools like Visual Studio Code, developers can easily deploy the website to Azure. The seamless integration between development and deployment tools ensures a smooth process, allowing for rapid updates and continuous improvement.

Cost-Effectiveness: Azure's pay-as-you-go model allows the project to optimize costs by only paying for the resources used. This flexibility makes it easier to manage budgets while still delivering a high-quality hosting environment.

The website hosting process involves configuring the Azure environment, deploying the web application, and setting up supporting services like databases and authentication mechanisms. This approach ensures a reliable and efficient hosting solution, allowing users to book flights with confidence and ease.

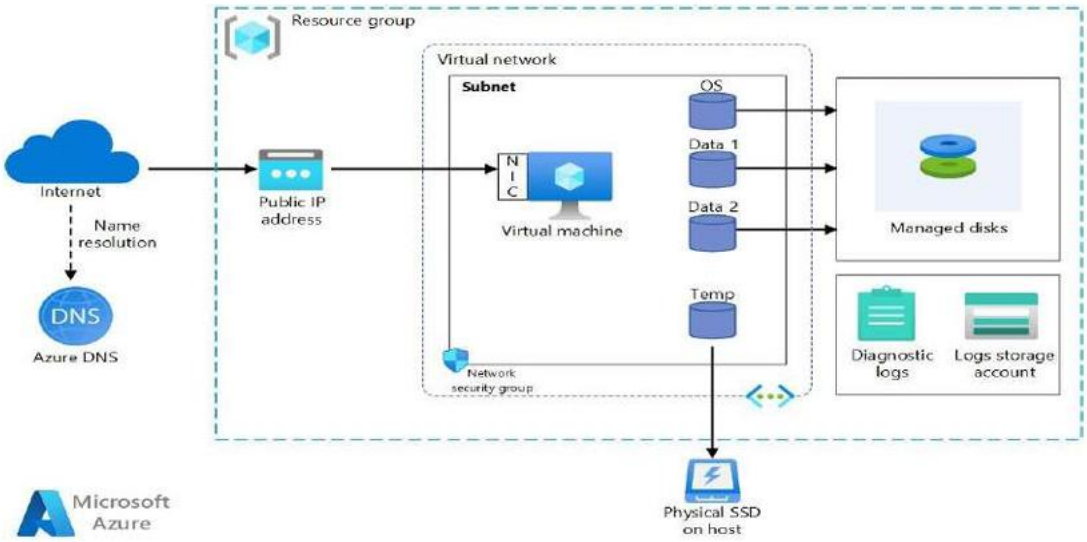
Overall, hosting the "Online Flight Booking System" on Microsoft Azure offers a scalable, secure, and high-performance platform. This enables the project to meet user demands while maintaining a strong focus on reliability and data protection.

PLATFORM

Visual Studio Code : Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

XAMPP: XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. We used HTML & CSS as Frontend & PHP as Backend.

Microsoft Azure: Microsoft Azure is a cloud computing platform & infrastructure created by Microsoft for building, deploying, and managing applications and services through a global network of Microsoft-managed data-centers. It provides a variety of services including virtual machines, web and mobile app development, data storage and management, and much more. Azure supports a range of programming languages, tools, & frameworks and integrates with a variety of third-party services.



IMPLEMENTATION STEPS FOR ONLINE FLIGHT BOOKING SYSTEM ON MICROSOFT AZURE:

DESIGN THE FLIGHT BOOKING SYSTEM

The first step in creating your online flight booking system is to design its overall structure and layout. This involves outlining the core components and deciding how each part of the system will interact with the others. Given that your system has three main roles—admin, staff, and user—you need to plan how each of these roles will function within the system.

For the user role, determine how the flight booking process will work. Consider what information users need to see when booking a flight, such as available flights, departure and arrival times, and other details. The staff role involves viewing tickets and checking seat availability. Plan the interface for staff so that they can easily access this information. The admin role requires a comprehensive view of the

system, including the ability to manage users, add new airlines, and update flight details. Design the admin interface to be intuitive, allowing for smooth management of the system.

DEVELOP THE WEB APPLICATION

Once the design is complete, the next step is to develop the web application. Using Visual Studio Code (VSCode), you can start building the website. This involves developing both the front-end and the back-end of the application. For the front end, you'll use HTML for structure, CSS for styling, and JavaScript for interactivity. Frameworks like Bootstrap can help create a responsive and visually appealing design.

On the back end, you'll work with PHP to handle server-side processing and MySQL to manage the database. The back-end is responsible for processing flight bookings, storing user data, and managing the flow of information between different parts of the system. This back-end development requires careful attention to data security and validation to ensure the system's reliability and safety.

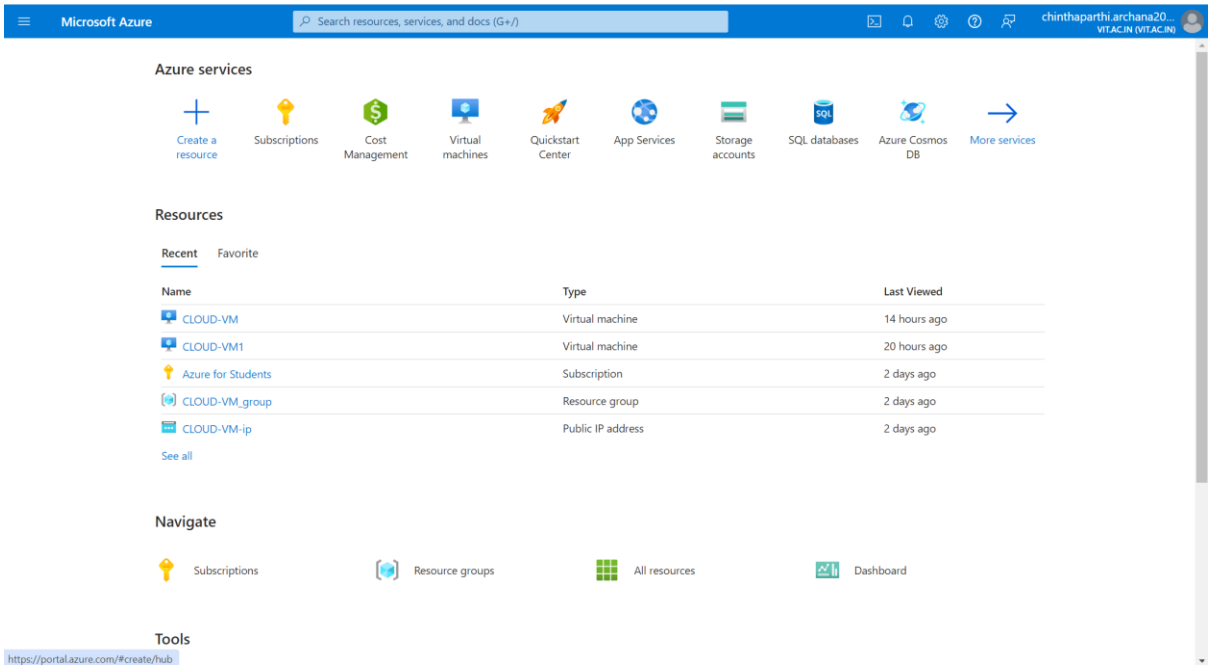
During the development phase, you may also use tools like Ajax for asynchronous data loading and JQuery for simplified JavaScript operations. This combination of technologies allows for a dynamic and responsive user experience. As you build the web application, make sure to test each feature thoroughly to ensure it works as expected and meets user requirements.

SET UP AZURE ENVIRONMENT

To start using Microsoft Azure, you'll first need to create an account. Visit the Microsoft Azure website and click on the sign-up option. Fill in the necessary details, including your email address and a secure password. Follow the prompts to complete the registration process.

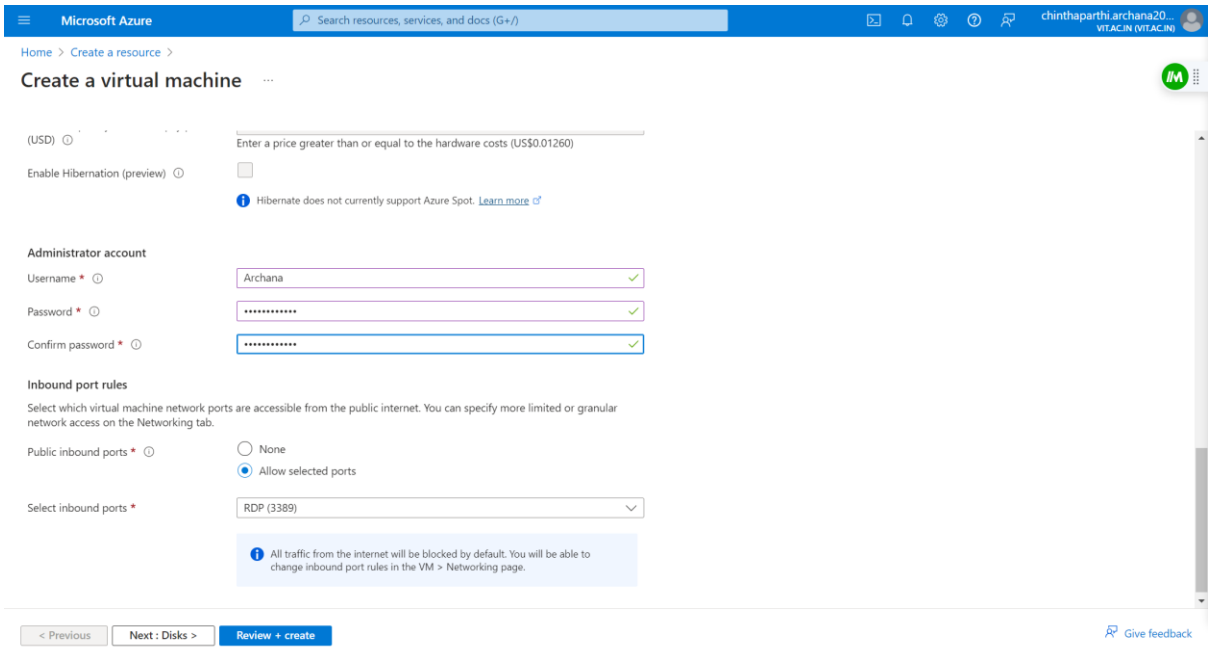
Once you've signed up, you can log in to the Azure portal, where you can explore different features and manage your account settings. The Azure portal is where you can access and set up your cloud-based resources.

MICROSOFT AZURE HOMEPAGE



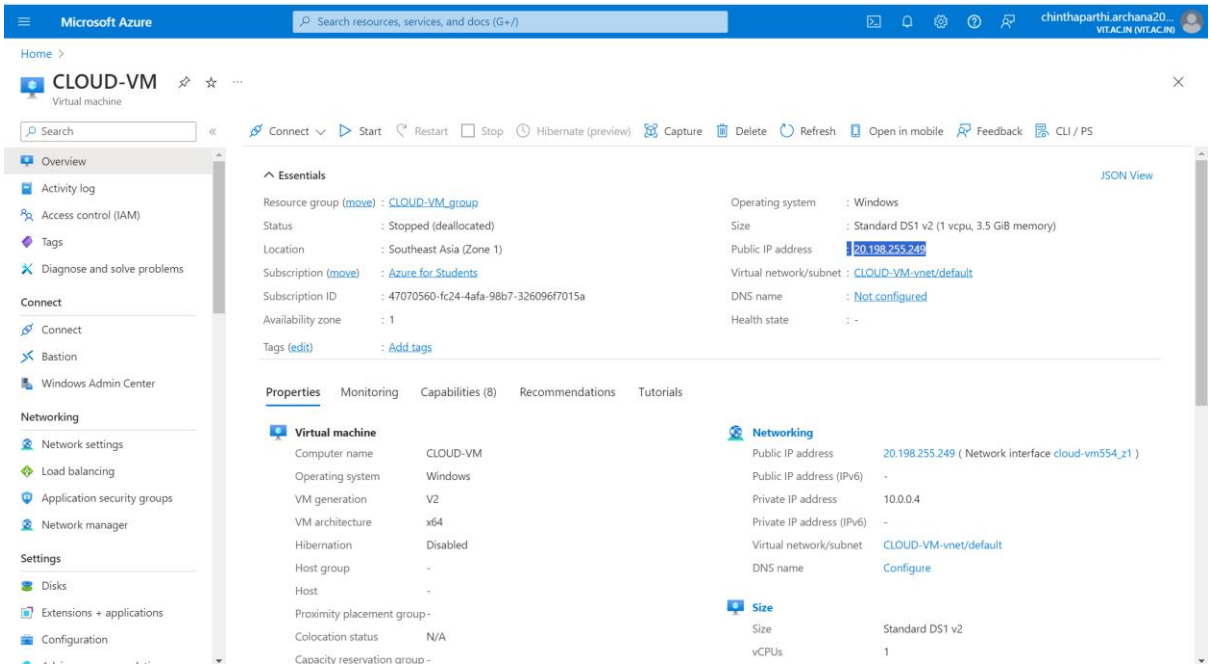
After logging in to Microsoft Azure, the next step is to create a virtual machine (VM) for your project. Start by clicking on "Create a resource." You will be prompted to enter project details such as the resource group, virtual machine name, region, security type, and operating system image. Fill out these details according to your project requirements.

CREATING A VIRTUAL MACHINE



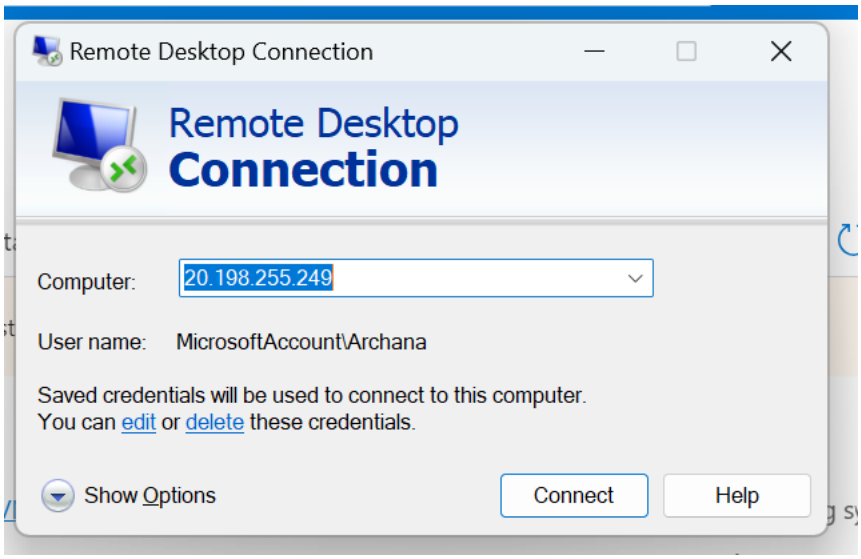
Additionally, you need to create a user account for the VM by entering a username and password. It's important to remember this information because it will be required for others to access your VM using the public IP address, username, and password. Once all the information is entered, click on "Review + create." After a brief review, your virtual machine will be created, and ready for you to use.

CLOUD VM DETAILS AND PUBLIC IP ADDRESS

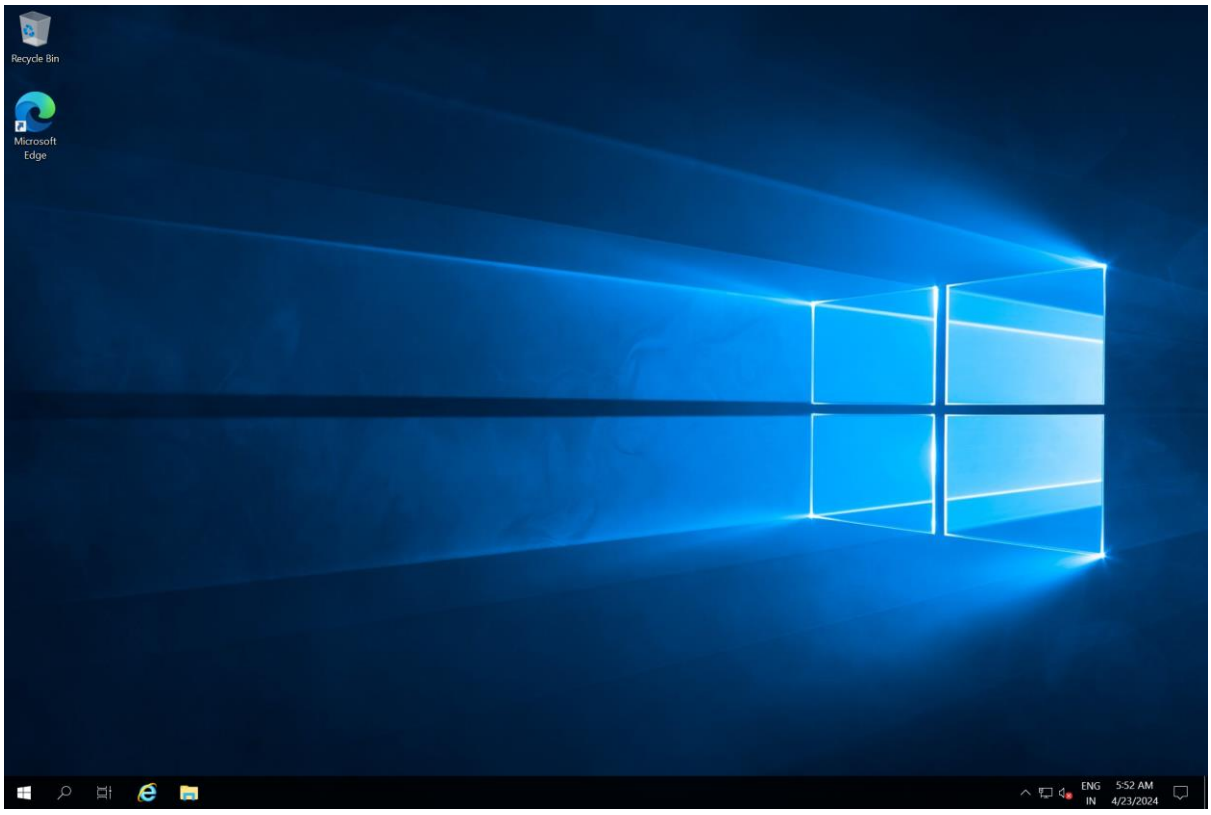


After creating the virtual machine (VM), navigate to the VM's details page in the Azure portal. Click on "Start" to begin the virtual machine's startup process, which might take about a minute. Once the VM is running, copy the public IP address displayed on the VM details page.

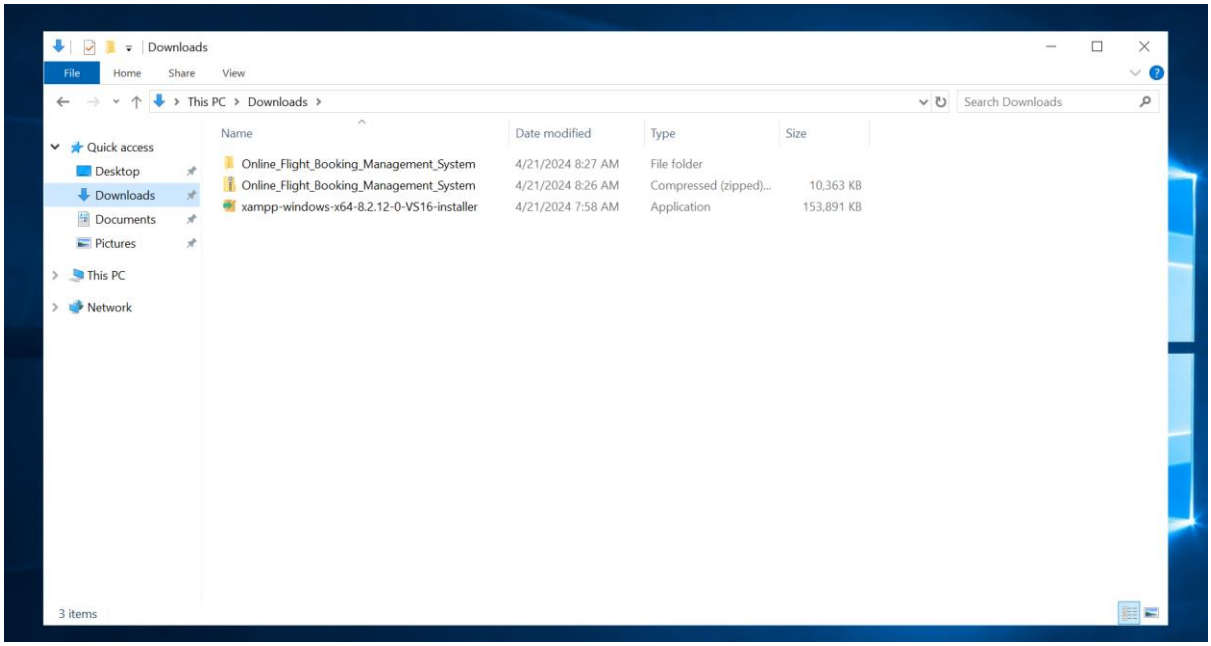
Open the Remote Desktop Connection on your computer and paste the copied public IP address in the connection box. Click on "Connect," and you'll be prompted to enter the VM's username and password. Ensure you enter the correct credentials, as any errors will prevent connection. If the information is correct, your remote desktop session will start, giving you full access to the virtual machine's environment.

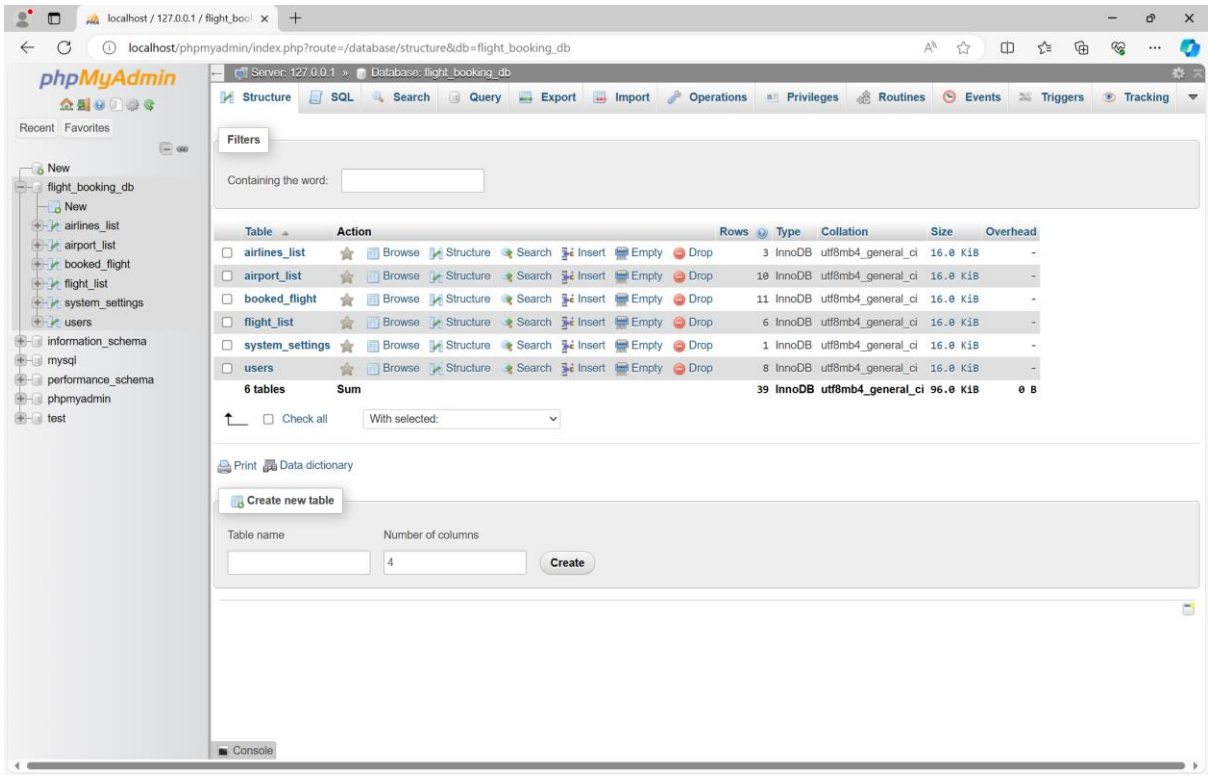


VIRTUAL MACHINE



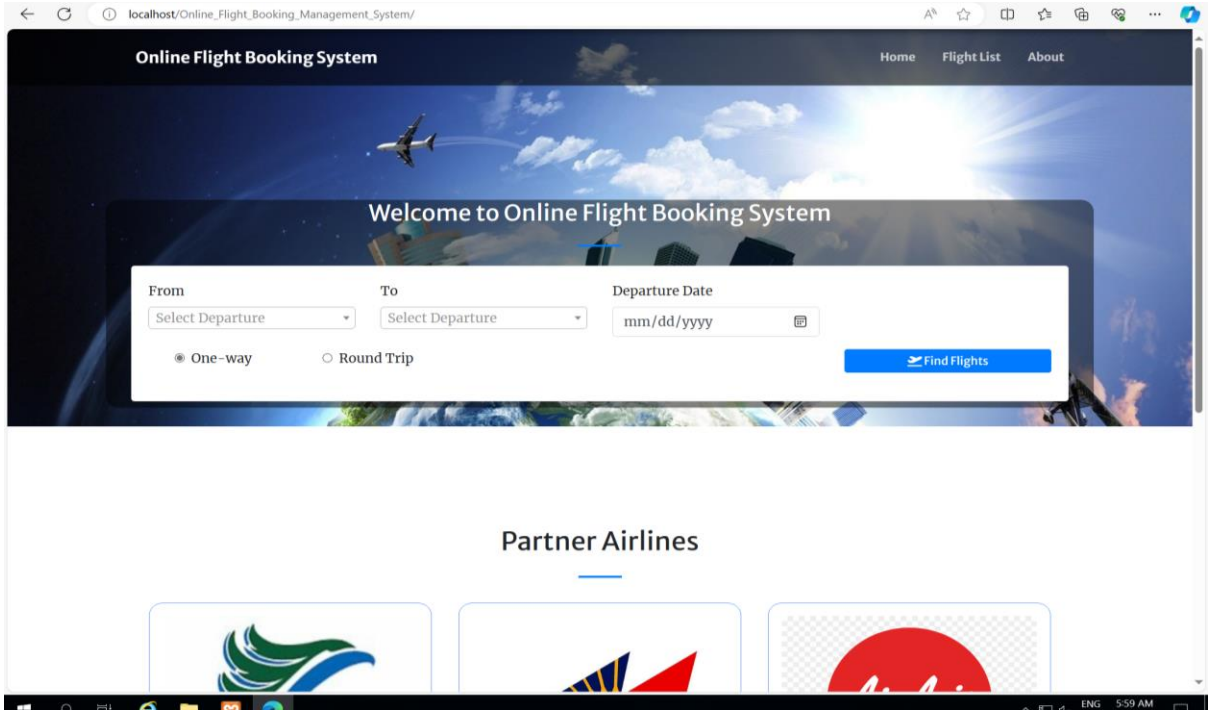
Once you have accessed your virtual machine (VM) through Remote Desktop Connection, you need to install the necessary software to run your project. Start by downloading and installing XAMPP, which provides Apache, MySQL, PHP, and Perl, creating a suitable local server environment. After installing XAMPP, open the control panel to start Apache and MySQL. Then, download your project source code from GitHub, either as a ZIP file or by cloning the repository. After extracting the files, set up your database in XAMPP by creating a new database in phpMyAdmin and importing the required schema. Finally, ensure your code is configured to connect to this database, using the correct credentials and database name. With these steps, your VM will be ready to support your online flight booking system.





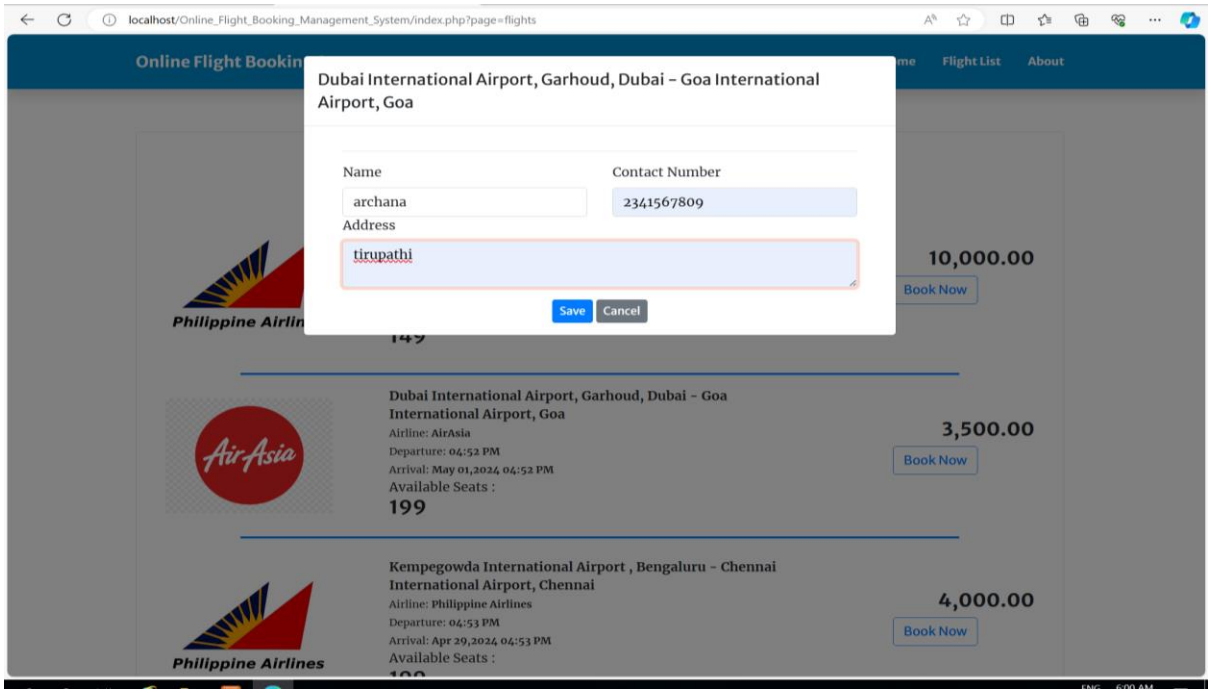
After installing and starting Apache and MySQL in XAMPP, you can test your online flight ticket booking system. Open any web browser and enter "http://localhost/Online_Flight_Booking_Management_System" in the address bar. This should take you to the homepage of your online flight ticket booking system, allowing you to verify that the website is running correctly on your virtual machine. If everything is set up properly, you should see the website's main interface, confirming that the server environment and code are functioning as expected.

HOME PAGE OF ONLINE FLIGHT BOOKING SYSTEM IN THE VIRTUAL MACHINE



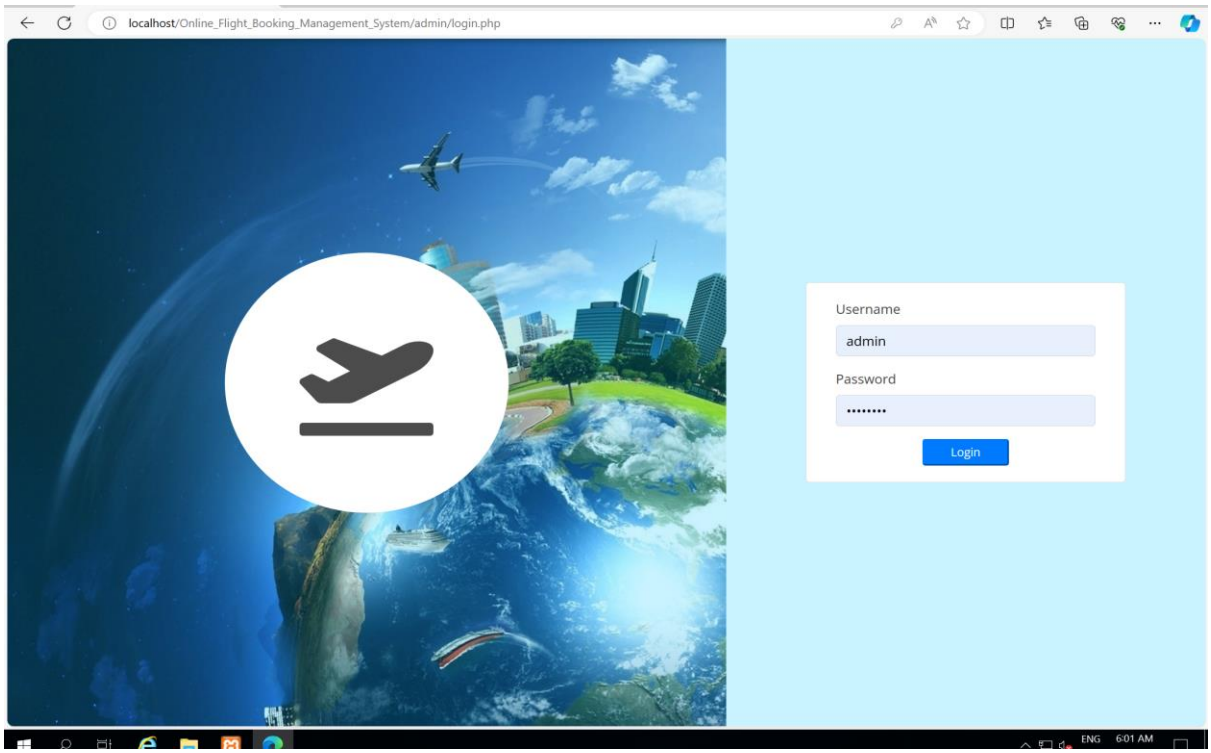
On the homepage of your online flight ticket booking system, you can search for flights by selecting the departure and arrival locations, along with your preferred travel date. Alternatively, you can click on the "Flight List" to see all available flights. To book a flight, simply select "Book Now" next to your chosen flight. You will then need to enter the number of seats, along with the traveler's name, contact number, and address. Once you've filled in the required details, click "Save." If the booking is successful, you'll see a pop-up message indicating, "Flight booked successfully." This confirmation helps users know that their booking has been completed.

TICKET BOOKING



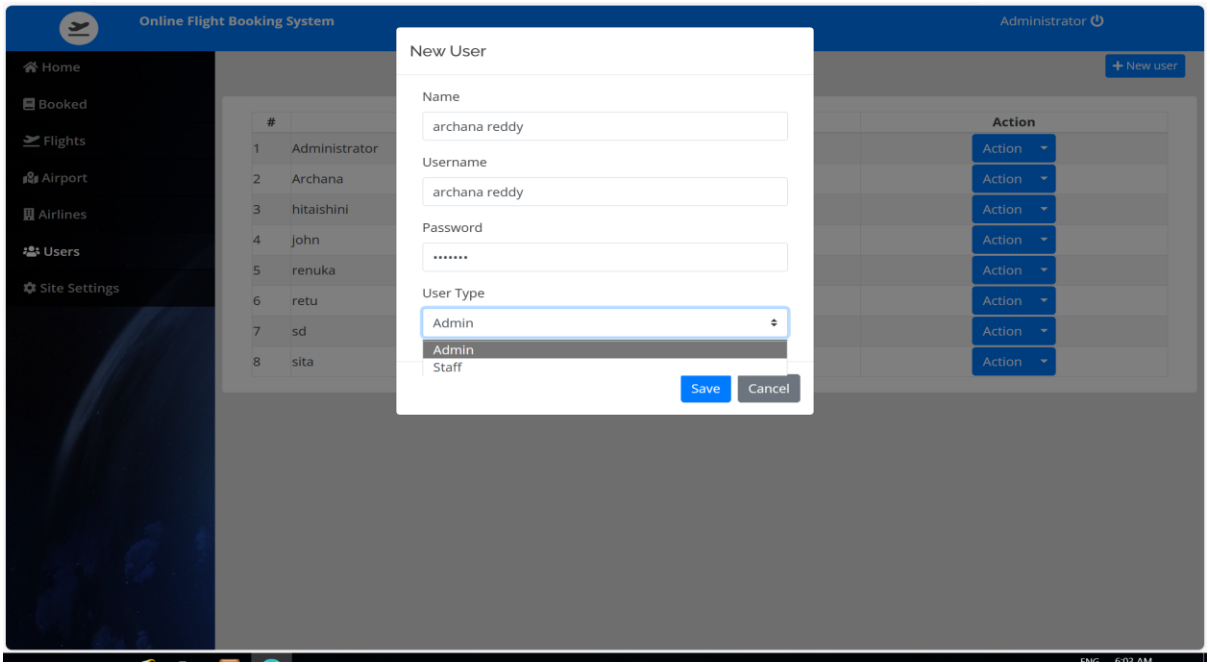
To access the administrative interface for your online flight booking system, open any internet browser and navigate to "http://localhost/Online_Flight_Booking_Management_System/admin/login.php." This will bring you to a login page where you need to enter your admin username and password. Once you input these credentials and click "Login," you will be directed to the admin home page, where you can manage various administrative tasks related to the flight booking system.

ADMIN LOGIN PAGE



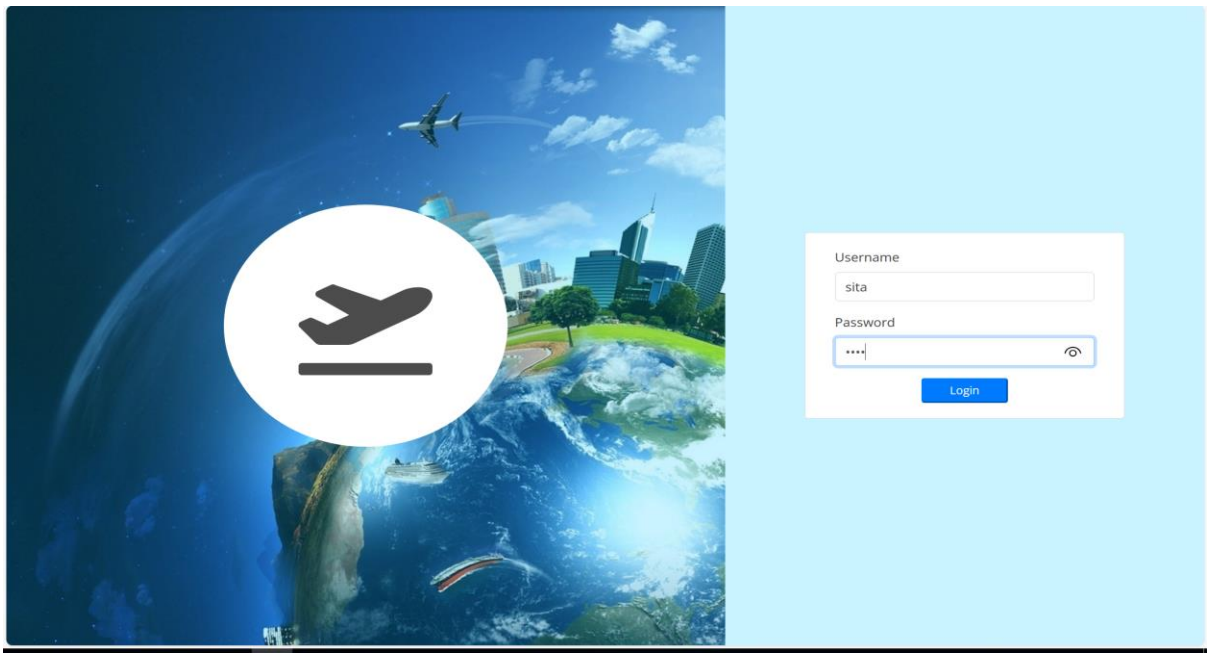
In the online flight booking system, the admin has comprehensive control over bookings, flights, and user management. They can view booked flights, check seat availability, and cancel flights if necessary. Additionally, admins can manage user accounts by adding new users with specific roles—either admin or staff. Admins can see all system details and have full privileges, while staff have limited access, primarily for viewing and booking flights without the ability to alter the system's structure or user roles. This robust administrative control ensures smooth operations and efficient management of the booking system

ADDING NEW USER



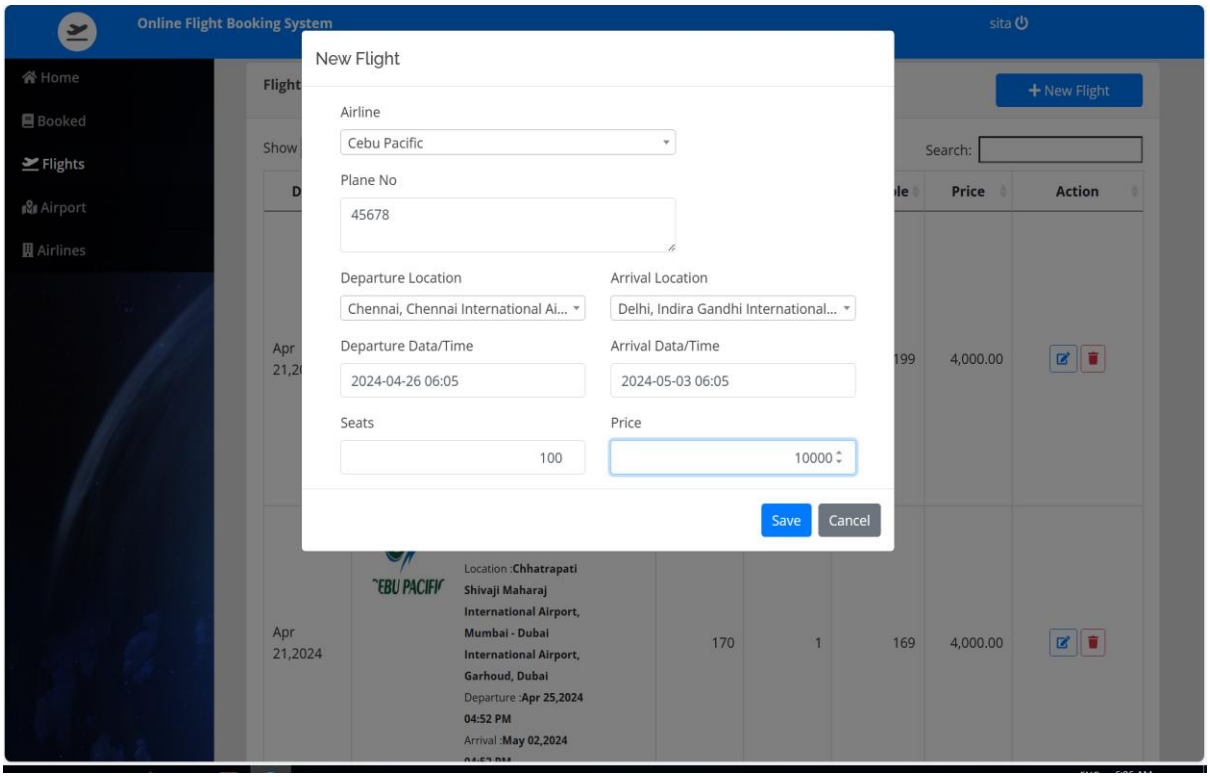
When you open any internet browser and visit "http://localhost/Online_Flight_Booking_Management_System/login.php", you'll find the login page for staff. Here, you need to enter your username and password to access the home page.

STAFF LOGIN PAGE



Staff members in the Online Flight Booking System have access to essential information like booked flights, available flights, seat availability, and airline details. They can also cancel booked flights in case of emergencies, providing flexibility and responsiveness. However, staff have less control compared to administrators; they cannot view user details, add or delete users, or access site settings. This reduced level of access ensures that critical administrative functions remain within the domain of the admin, while staff can focus on day-to-day operations and assisting customers with their flight bookings.

ADDING NEW FLIGHT



CODE

book_flight.php

```
<div class="container-fluid">
        <div class="col-lg-12">
        <form action="" id="book-flight">
                <input type="hidden" name="flight_id" value="<?php echo $_GET['id'] ?>">
                <div class="form-group row" id="qty">
                        <div class="col-md-3">
                        <label for="" class="control-label">Person/s</label>
                        <input type="number" class="form-control text-right" min='1' value="1" id="count" max="<?php echo $_GET['max'] ?>">
                        </div>
```

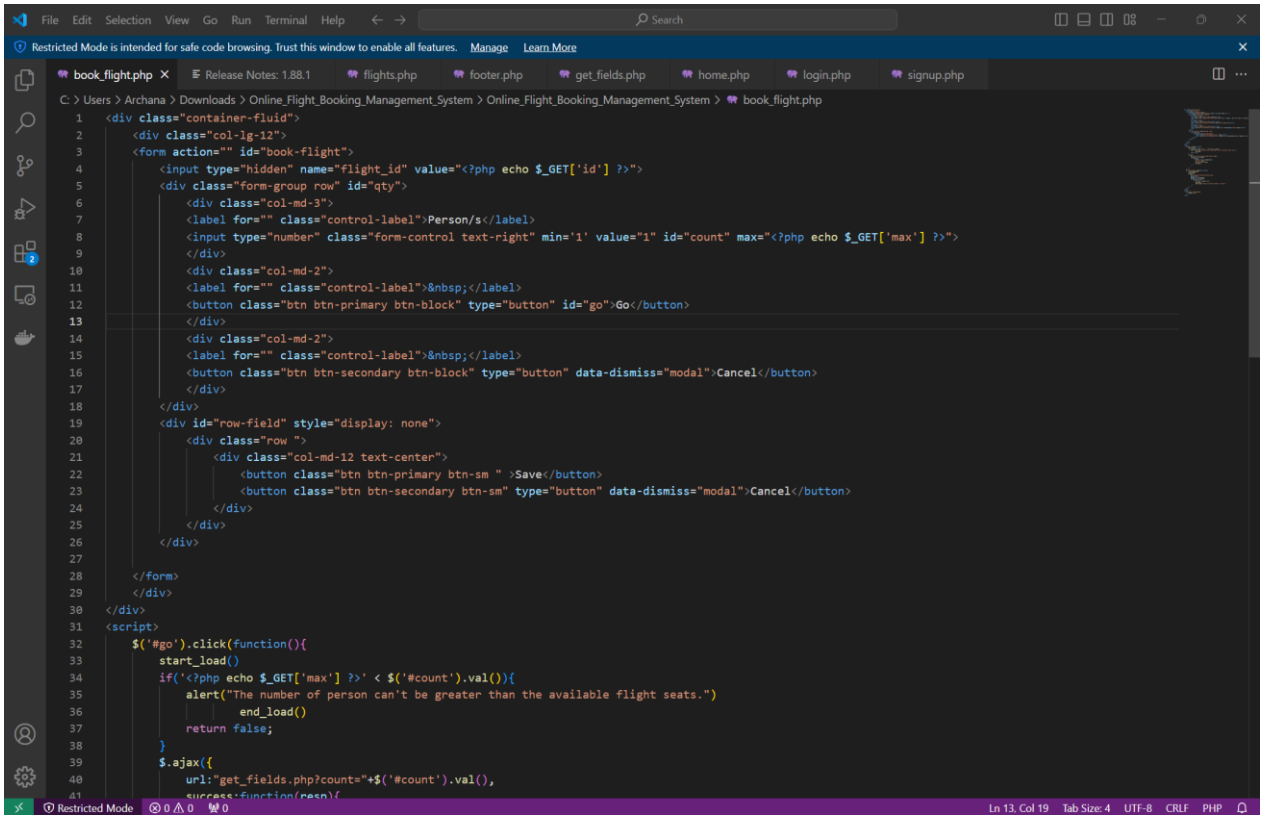
```
                        <div class="col-md-2">
                        <label for="" class="control-label"> </label>
                        <button class="btn btn-primary btn-block" type="button" id="go">Go</button>
                        </div>
                        <div class="col-md-2">
                        <label for="" class="control-label"> </label>
                        <button class="btn btn-secondary btn-block" type="button" data-dismiss="modal">Cancel</button>
                        </div>
                </div>
                <div id="row-field" style="display: none">
                        <div class="row ">
                                <div class="col-md-12 text-center">
                                        <button class="btn btn-primary btn-sm " >Save</button>
                                        <button class="btn btn-secondary btn-sm" type="button" data-dismiss="modal">Cancel</button>
                                </div>
                        </div>
                </div>

        </form>
        </div>
</div>
<script>
        $('#go').click(function(){
                start_load()
                if('<?php echo $_GET['max'] ?>' < $('#count').val()){
                        alert("The number of person can't be greater than the available flight seats.")
                                end_load()
                        return false;
                }
```

```
		$.ajax({
			url:"get_fields.php?count="+$('#count').val(),
			success:function(resp){
				if(resp){
					$('#row-field').prepend(resp)
					$('#qty').hide()
					$('#row-field').show()
					end_load()
				}
			}

		})
	})
	$('#book-flight').submit(function(e){
		e.preventDefault()
		start_load()
		$.ajax({
			url:'admin/ajax.php?action=book_flight',
			method:"POST",
			data:$(this).serialize(),
			success:function(resp){
				if(resp ==1 ){
					$('.modal').modal('hide')
					end_load()
					alert_toast("Flight successfully booked.","success")
				}
			}
		})
	})
</script>
<style>
```

```
        #uni_modal .modal-footer{
                display: none
        }
</style>
```



TEST CASES:

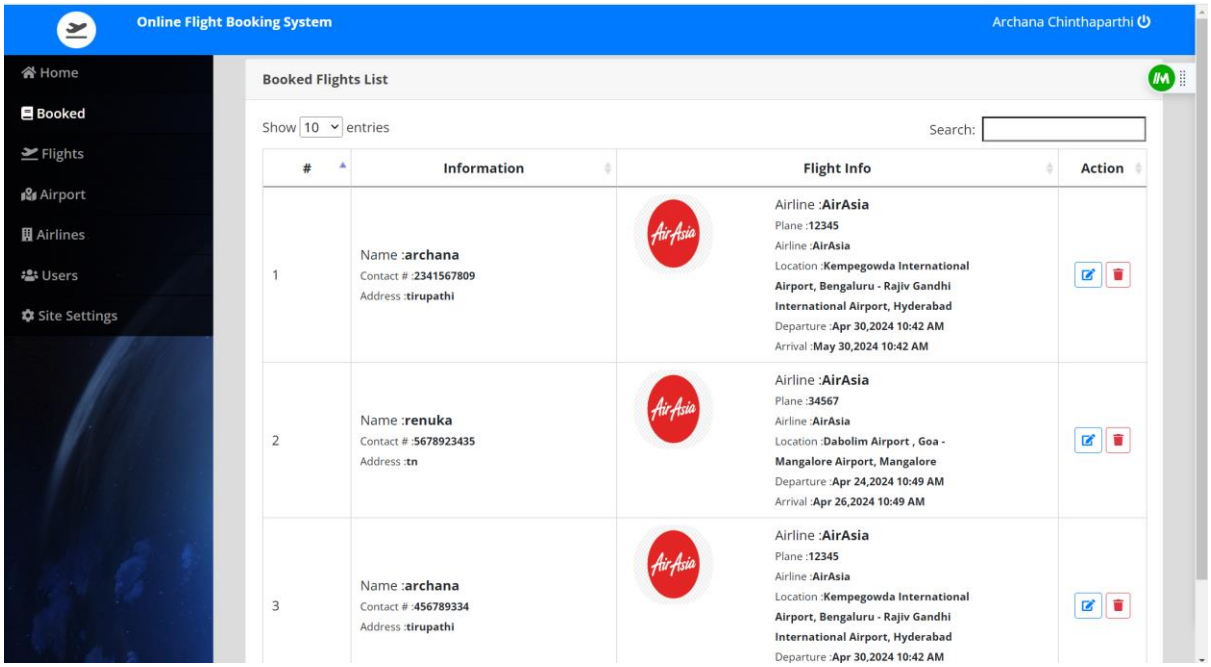
Test Case ID	Description	Expected Outcome	Actual Outcome	Status
TC01	Admin enters correct username and password	The admin panel should open	Admin panel opened	Passed
TC02	Staff enters the correct username but the wrong password	Should not allow access and show error	Access denied, error message shown	Passed
TC03	Remote Desktop with incorrect public IP address	Should not connect, show error message	Connection error message shown	Passed
TC04	Remote Desktop with correct IP, but wrong password	Should not connect, show error message	Connection error message shown	Passed

OUTCOME OF PROJECT

The "Online Flight Ticket Booking System" project relies on a cloud-based approach to ensure scalability, security, and high availability. The theoretical foundation for this methodology is rooted in the principles of cloud computing, which enable distributed resources, flexible scaling, and enhanced security.

By deploying the system on Microsoft Azure, the project leverages cloud infrastructure to deliver a robust, reliable, and scalable solution. This approach allows the system to efficiently handle varying user demands, maintain high performance during peak times, and ensure data security through Azure's built-in security features. The theory guiding this solution is based on the advantages of cloud computing in providing scalable resources and secure, resilient systems.

Users can successfully book flight tickets using the "Online Flight Ticket Booking System," deployed on Microsoft Azure, thanks to the intuitive user interface and straightforward booking process. After selecting the departure and arrival dates, users can easily choose from available flights and confirm their booking with a few clicks. This seamless booking experience is designed to minimize user effort while ensuring accurate data entry, contributing to an efficient and reliable flight booking system hosted on Azure's robust cloud platform.



FEATURES UTILIZED IN MICROSOFT AZURE FOR ONLINE FLIGHT TICKET BOOKING SYSTEM

The implementation of the "Online Flight Ticket Booking System" leverages key features of Microsoft Azure to provide a robust, scalable, and secure platform. Here's an overview of the primary Azure features used in this project:

Azure App Service:

The project is deployed on Azure App Service, a fully managed platform that offers scalability, high availability, and easy integration with other Azure services. This flexibility allows the system to scale dynamically based on user demand, ensuring a consistent user experience even during peak times.

Azure SQL Database:

To store essential data like flight schedules, booking information, and user accounts, the system uses Azure SQL Database. This managed database service provides automated backups, disaster recovery options, and high security, ensuring that your data remains safe and compliant with industry standards.

Azure Virtual Machines:

The project also utilizes Azure Virtual Machines to run and test the application. This feature allows for custom configurations and provides a controlled environment for hosting additional services if needed. It offers a balance of flexibility and scalability for specific project requirements.

Azure Active Directory:

User authentication and role-based access control are managed through Azure Active Directory. This service enables secure login, allowing administrators to control user permissions and ensure that only authorized users can access specific parts of the system.

Azure Resource Group:

The project is organized within an Azure Resource Group, providing a way to manage and maintain all the related Azure resources in one place. This setup simplifies deployment, monitoring, and resource management, allowing for easier scaling and maintenance.

Together, these Azure features create a strong foundation for the Online Flight Ticket Booking System, ensuring a seamless user experience, high security, and easy scalability.

LIMITATIONS

Using Microsoft Azure for student projects has certain limitations, particularly when it comes to cost and resource constraints. Here are some key limitations that might affect students working on projects like an Online Flight Ticket Booking System:

Free Credits and Budget Constraints:

Azure offers free credits to students for a limited time, but once these credits are exhausted, students must purchase additional resources, which can be costly. This limitation might hinder ongoing development, testing, and scaling, as students may not have the budget to continue using paid Azure services.

Limited Resource Quotas:

Azure provides limited resources within the free tier, such as a maximum number of virtual machines or storage capacity. If a project requires more resources, students must upgrade to a paid plan, adding to the overall cost.

Complexity and Learning Curve:

Microsoft Azure offers a broad range of services, each with its own learning curve. Students may find it challenging to master the complexity of Azure, especially when integrating multiple services into a project. This limitation can slow down development and implementation.

Geographic Limitations:

Azure services are distributed across various regions, but some services may not be available in specific locations. This limitation can impact students working on projects in regions where certain Azure services are restricted or unavailable.

Limitations in Online Flight Ticket Booking System

From the perspective of the website, the following limitations are observed:

No Seat Position Information:

The website does not provide details about specific seat positions, such as window or aisle seats. This limitation can affect user experience, as customers may prefer to choose their seats based on location.

Restricted User Privileges:

Users with limited privileges, such as staff members, cannot view seat layouts or other detailed information about booked flights. This limitation restricts staff from accessing specific data, which might affect their ability to assist users effectively.

Together, these limitations highlight the potential challenges for students working with Microsoft Azure and the Online Flight Ticket Booking System. Addressing these limitations might require additional resources, expertise, or a more flexible approach to system design and user interaction.

CONCLUSION

The Online Flight Ticket Booking System was successfully deployed on Microsoft Azure, providing a robust platform for booking flights. The deployment process involved setting up an Azure environment and configuring the necessary services to ensure high performance and scalability. The system allows users to search for flights, book tickets, and manage their accounts, with administrators having additional controls to oversee the system's operations.

By using Azure App Service for deployment, the system benefits from a flexible, scalable environment that can handle varying levels of traffic. Azure SQL Database serves as the backend for storing booking information and user data, offering automated backups and high security. The successful deployment and operation of this system demonstrate the power of cloud-based solutions for managing online services, offering a reliable and secure experience for users and administrators alike.

REFERENCES :

1. Kaur, G., & Kaur, R. (2018). Architecture Model for Flight Booking System Based on Web Services. International Journal of Advanced Research in Computer Science and Software Engineering, 8(1), 104-108.
2. Korin, A. (2023, April 17). NASA Funds Experimentation With Cloud-Based Flight Management System. Aviation Week.
3. Bhardwaj, P., & Sharma, M. (2019). Implementing a Cloud-Based Flight Management System. International Journal of Computer Applications, 178(1), 1-5.
4. Wiseman, Y. (2016). Can Flight Data Recorder Memory Be Stored on the Cloud? Journal of Aviation Technology and Engineering, 6(1), Article 3.
5. Borodulkin, V., & Kholod, A. (2021). Demonstrating a Cloud-Enabled Flight Management System. Journal of Aerospace Information Systems, 18(11), 815-826.
6. Al-Hamadi, A., Al-Bayati, B., & Al-Nuaimi, M. (2021). A cryptographic cloud-based approach for the mitigation of the airline cargo cancellation problem. Journal of Intelligent Transportation Systems, 25(6), 548-561.
7. Kumar, V., & Kaur, G. (2021). Improved wild horse optimization with levy flight algorithm for effective task scheduling in cloud computing. Journal of Intelligent & Fuzzy Systems, 41(1), 1057-1068.
8. Li, X., Zhang, Y., & Wang, H. (2021). Cloud-based flight delay prediction using logistic regression. Journal of Intelligent Transportation Systems, 25(3), 245-256.
9. Sharma, V., & Kumar, A.(2023)Automating the Indian transportation system through intelligent searching and retrieving with Amazon Elastic Compute Cloud. Journal of Cloud Computing, 12(2).
10. Pawel Kruk S .(2017). A formal model for the interoperability of service clouds. Journal of Network and Computer Applications, 99, 1-15.