Building Cloud Native Applications with Ballerina and Choreo

Training Objective:

In this training, you will:

1. Build an HTTP service(API) using Ballerina.
2. Use Choreo to deploy the cloud-native application.
    a. Deploy a Ballerina service.
    b. Deploy a web application.

High-Level Steps:

1. Develop the HTTP service using the Ballerina programming language.
2. Push the code to your GitHub account.
3. Deploy the cloud-native application on Choreo.
    a. Deploy the Ballerina HTTP service
    b. Deploy the ReactJS web application

Prerequisites:

1. A GitHub Account.
2. Git installed in your workstation.
3. A recent version of Google Chrome, Mozilla Firefox.
4. Ballerina v 2201.8.4 installed in your workstation.
5. Microsoft Visual Studio (VSCode) and WSO2 Ballerina plugin
6. PostMan and curl (or any HTTP client) installed in your workstation.
7. Choreo Account

Business Scenario:

Build a reservation system for a Luxury Hotel.

Solution

Build a web application that enables users to book rooms. This application will offer the following features:

Room Search:

- Users can search for rooms by specifying check-in and check-out dates, with an option to filter by number of guests.
- Search results will showcase a list of room types (eg: single, double etc).
- Each room listing will feature a "Reserve" button for easy booking.

Room Reservation:

- To reserve a room, users need to input personal information: full name, contact number, and email.
- The form validates and activates the "Reserve" button once all fields are properly filled. Upon reservation, a unique reference number is displayed for the user to copy.

List My Reservations:

- Users can look up their reservations once they are logged in.
- Each entry in the list will provide options to either update the reservation details or cancel the reservation.

Update Reservations:

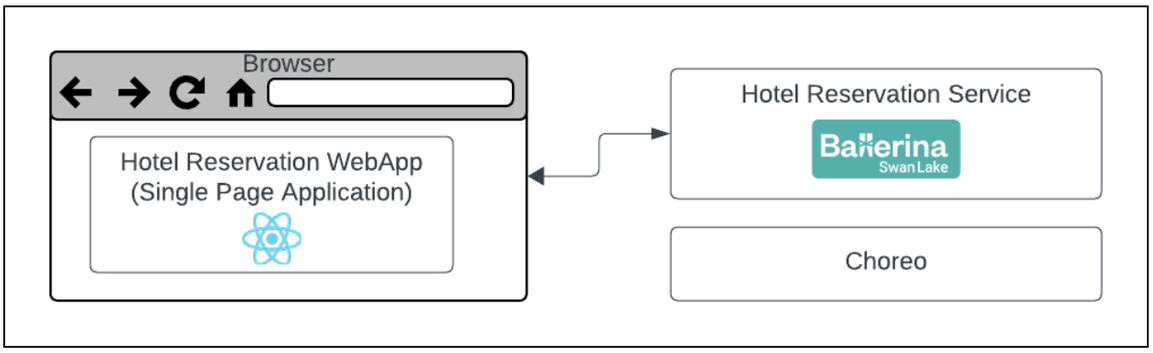
- Users have the flexibility to modify any part of their reservation.

Cancel Reservations:

- Users can cancel their reservations at any time, with a straightforward option to cancel their booking.

Design

The Hotel Reservation System will include a Hotel Reservation Service developed using Ballerina, and a Hotel Reservation Web Application developed using ReactJS.



Detailed Steps:

Developing the Hotel Reservation Service using Ballerina

The Hotel Reservation Service consists of a Ballerina HTTP service with 5 resources. A dedicated Ballerina project for the hotel reservation service has been created, complete with the required types and utility functions. During this session, we will focus on creating a Ballerina service and complete the implementation of the resources.

Base URL

http://localhost:9090/reservations

Resources

Resource	Path	Action	Query Param	Path Param	Request	Response

Get All available room types	/roomTypes/	GET	string checkinDate
			
			string checkoutDate
			
			int guestCapacity
						[
						{
						"id": 0,
						"name": "Single",
						"guestCapacity": 1,
						"price": 80
						
						},
						
						{
						"id": 0,
						"name": "Double",
						"guestCapacity": 2,
						"price": 100
						}

						]

Create a new reservation		POST			{
					"checkinDate": "2024-02-19T14:00:00Z",
					"checkoutDate": "2024-02-20T10:00:00Z",
					"rate": 100,
					"user": {
					"id": "123",
					"name": "waruna",
					"email": "waruna@someemail.com",
					"mobileNumber": "987"
					},
					"roomType": "Family"
					}
						{
						"id": "1",
						"checkinDate": "2024-02-19T14:00:00Z",
						"checkoutDate": "2024-02-20T10:00:00Z",
						"user": {
						"id": "123",
						"name": "waruna",
						"email": "waruna@someemail.com",
						"mobileNumber": "987"
						},
						"room": {
						"number" : 201,
						"type": {
						"id": 0,
						"name": "Double",
						"guestCapacity": 2,
						"price": 100
						}
						}
						}

Update an existing reservation		PUT		reservation_id	{
					"checkinDate": "2024-02-20T14:00:00Z",
					"checkoutDate": "2024-02-21T10:00:00Z"
					}
						{
						"id": "1",
						"checkinDate": "2024-02-19T14:00:00Z",
						"checkoutDate": "2024-02-21T10:00:00Z",
						"user": {
						"id": "123",
						"name": "waruna",
						"email": "waruna@someemail.com",
						"mobileNumber": "987"
						},
						"room": {
						"number" : 201,
						"type": {
						"id": 0,
						"name": "Double",
						"guestCapacity": 2,
						"price": 100
						}
						}
						}

Remove a reservation		DELTE		reservation_id		

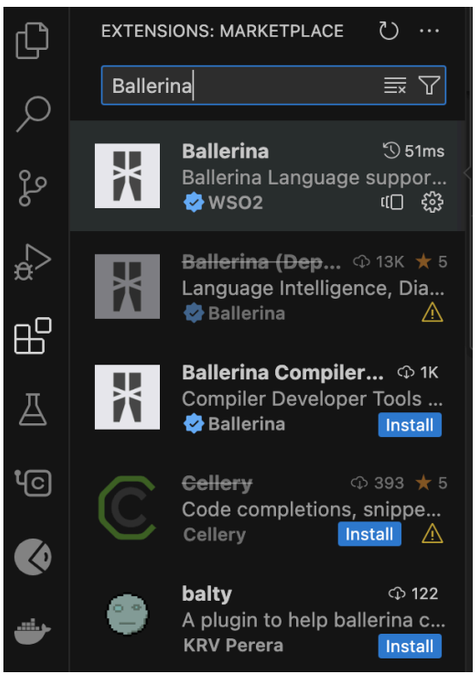
Get all reservations for a given user id	/users/	GET		userID		[
						{
						"checkinDate": "2024-02-19T14:00:00Z",
						"checkoutDate": "2024-02-20T10:00:00Z",
						"rate": 120,
						"user": {
						"id": "123",
						"name": "waruna",
						"email": "waruna@someemail.com",
						"mobileNumber": "987"
						},
						"roomType": "Family"
						},
						{
						"checkinDate": "2024-02-23T14:00:00Z",
						"checkoutDate": "2024-02-24T10:00:00Z",
						"rate": 100,
						"user": {
						"id": "123",
						"name": "waruna",
						"email": "waruna@someemail.com",
						"mobileNumber": "987"
						},
						"roomType": "Double"
						}
						]

Table 1: Reservation Service Table

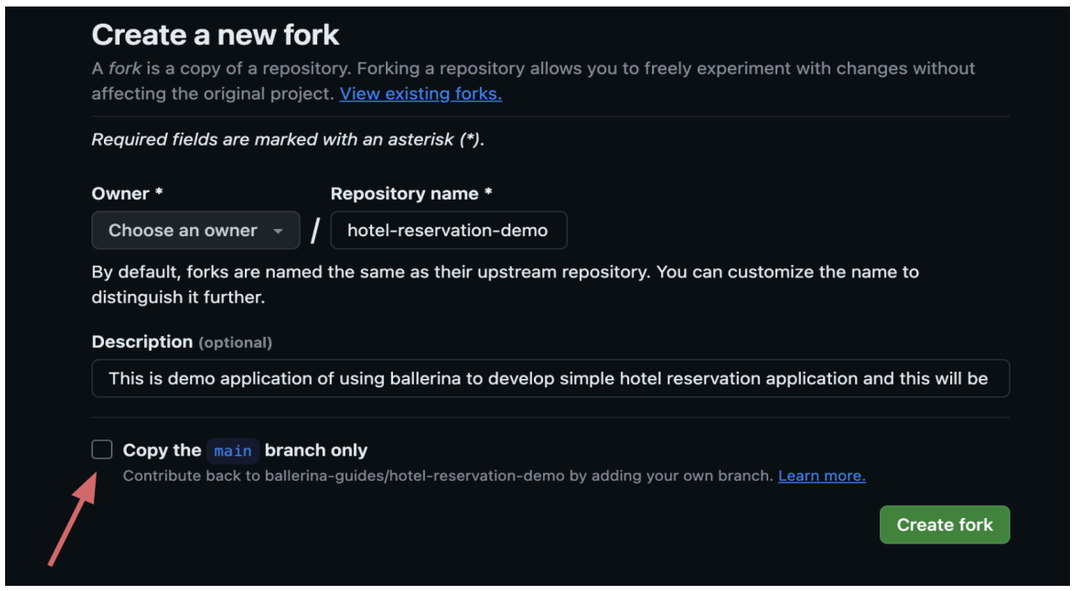
Step 1: Complete Post Request

1. Download and install Ballerina from the download page.

2. Install Ballerina Extension from Visual Studio Code



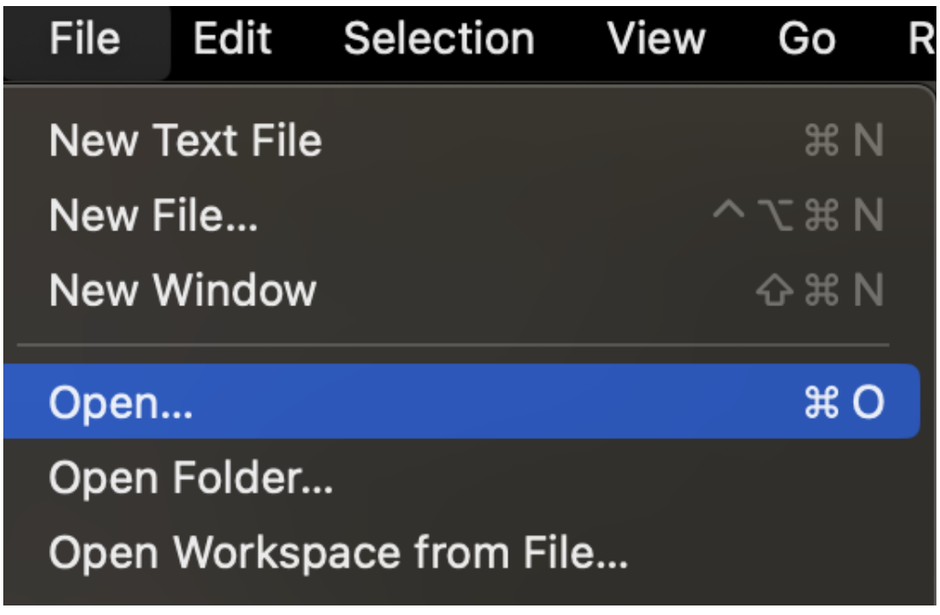
3. Fork the GitHub Repo - https://github.com/ballerina-guides/hotel-reservation-demo.
   Important: Make sure you untick the option “Copy the main branch only”.



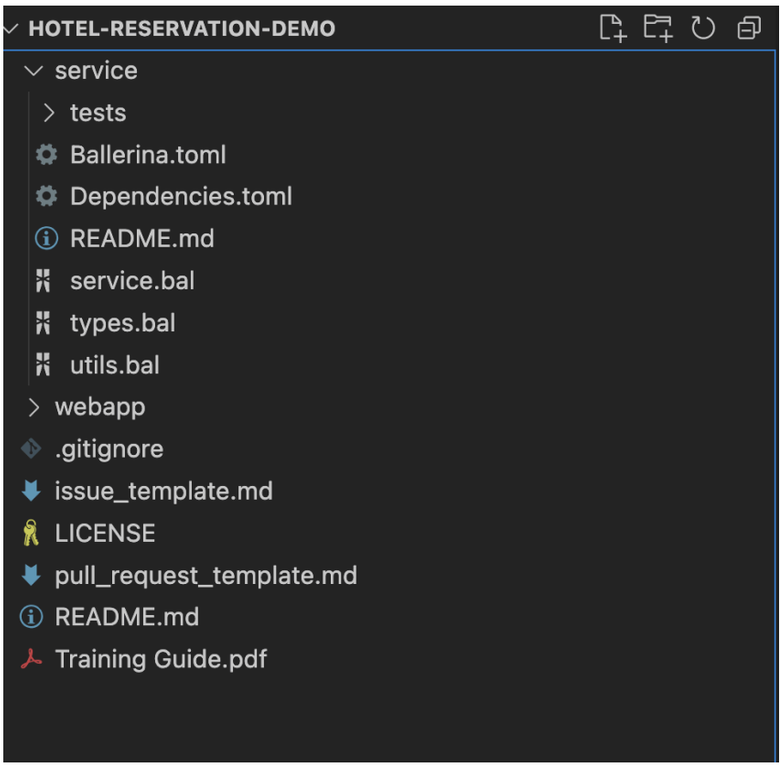
4. Open the Terminal (Command prompt Windows) and Clone the GitHub Repo https://github.com/<your_username>/ballerina-guides/hotel-reservation-demo using the following command.

```
Unset
git clone https://github.com/<your_username>/ballerina-guides/hotel-reservation-demo
```

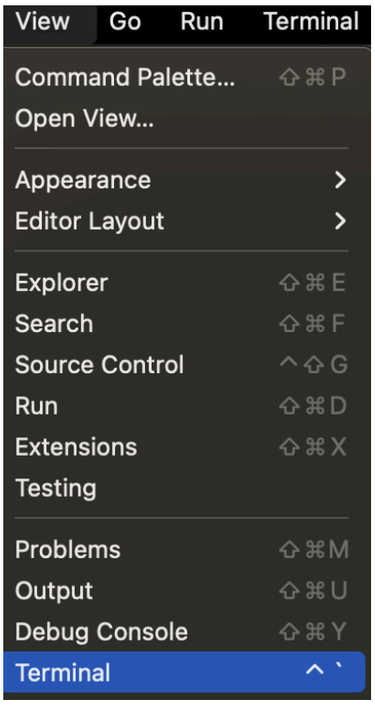
5. Open the hotel-reservation-demo directory using Visual Studio Code. First Click on **File > Open** and then select the hotel-reservation-demo folder and click **Open**.



6. You can see the following directory structure.



7. Now Click on View -> Terminal



8. Type following command in the Terminal and click Enter to run the command.

```
Unset
cd service
```

9. Now run the following command in the terminal to fix compilation errors in the code. This will download the relevant ballerina modules and fix the compilation issues.

```
Unset
bal build
```

10. If you still see any compilation errors, close the VScode and reopen it.

11. Now complete the body of the POST resource function the implementation to check whether a room is available for the given dates.You can use getAvailableRoom function to check whether a room is available and create a new reservation if room is available

12. Open Config.toml file in tests directory.

```
Unset
[user]
id="123"
name="waruna"
email=""
mobileNumber=""
```

Now change the name, mobile number and email to your name, mobile number and email.
Ex:

```
[user]
id="123"
name="John"
email="abc@gmail.com"
mobileNumber="+91123456789"
```

13. Run the following command in the Vscode terminal.

```
Unset
bal test
```

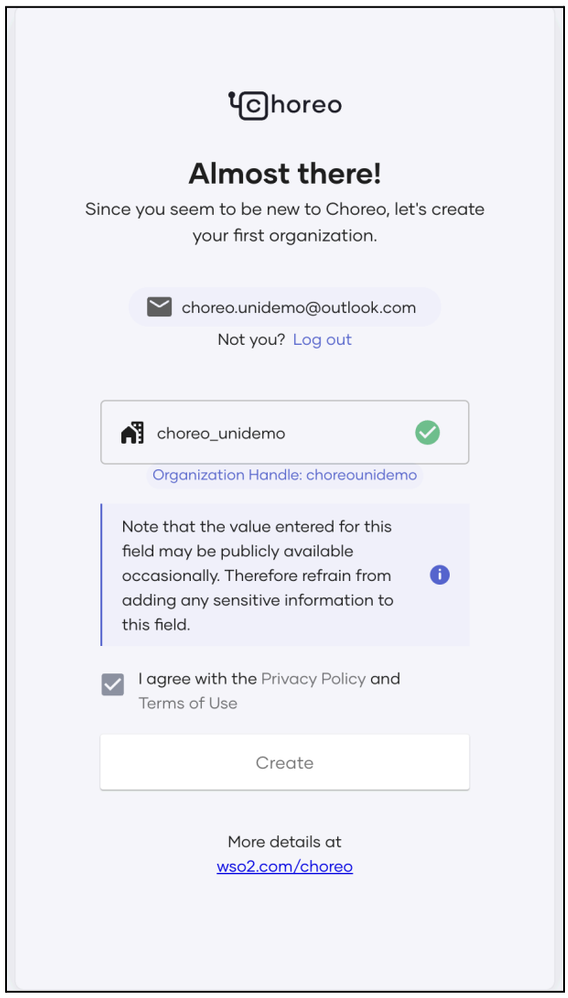
14. You will receive an email with a SMS as shown below:

Deploying the Hotel Reservation App using Choreo

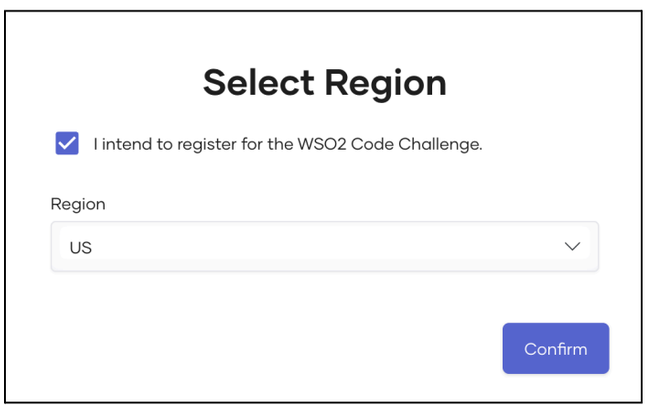
Step 1: Sign Up and Login to Choreo

1. Sign Up to Choreo by visiting the https://choreo.dev/ URL.

2. Once you log in to Choreo for the first time, you will be asked to provide an organization handle name. Provide a handle name and click **Create**.



3. In the region selector window, tick the **I intend to register for the WSO2 Code Challenge** and keep the default selected option of **US**. Then click **Confirm.**



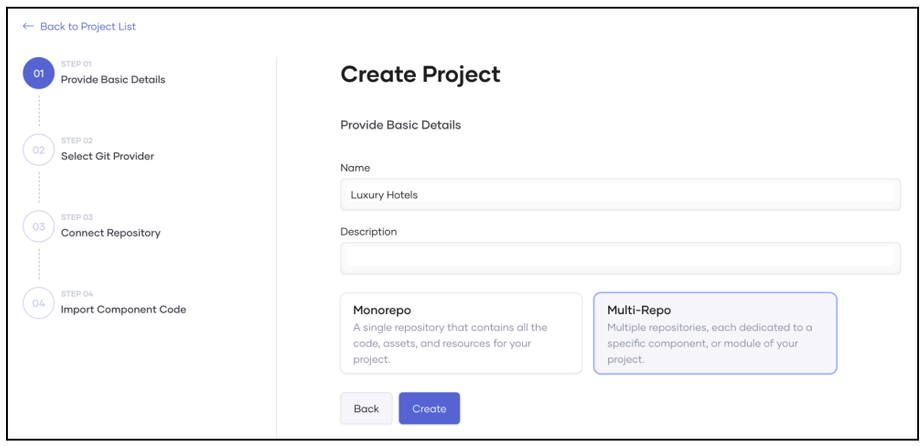
Step 2: Create a new Project

1. Click on the **Organization** card in the top menu.

2. Click the **Create Project** card.

3. Add the details shown below and click **Create.**

Field Name	Field Value
Name	Luxury Hotels
Project Type	Multi Repository



4. You will be directed to the overview of the project.

Step 3: Create and Deploy a Service for the Hotel Reservation Ballerina App

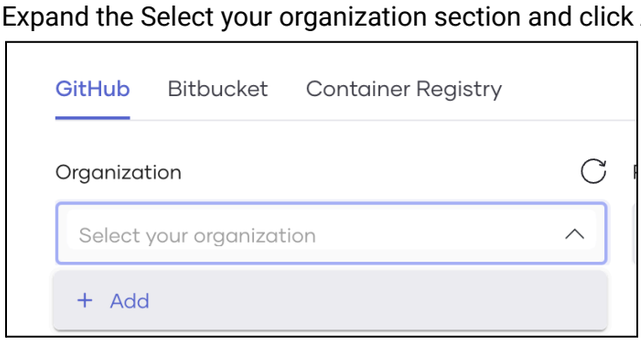
1. Select the **Service** card.

2. Provide a name for the Service.

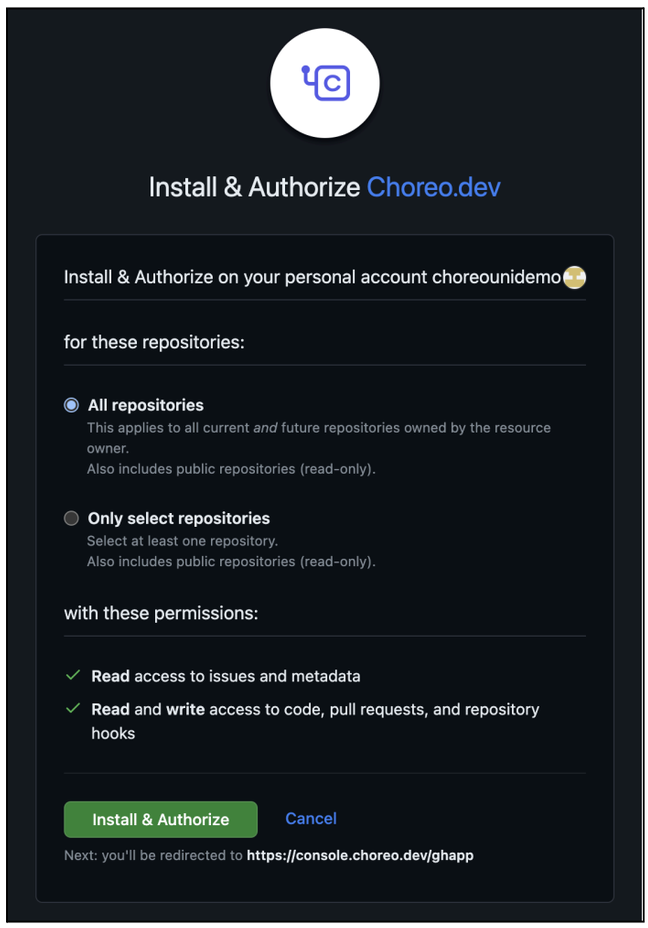
Field Name	Field Value
Name	Hotel Reservation Service

3. Click on the **Authorize With GitHub** and Click **Authorize Choreo.dev**.

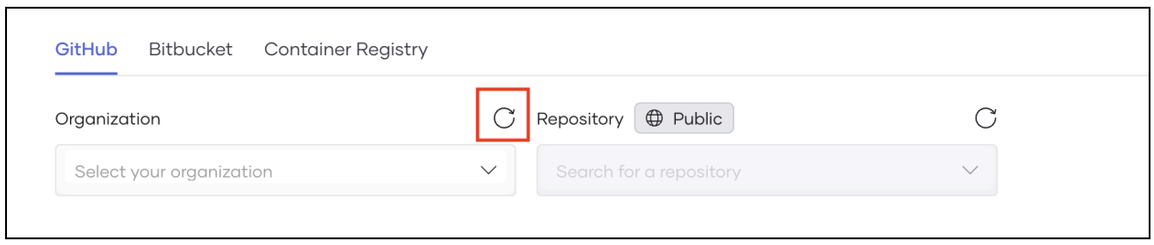
4. Expand the Select your organization section and click **Add**.



5. Keep the default selected option of **All repositories**. Then click **Install & Authorize**.



6. Click the refresh icon next to Organization.



7. Select the **hotel-reservation-demo** repository and select the branch **complete**.

8. Select **Ballerina** as the **Buildpack** and set the Ballerina Project Directory as below. Then select **Create**.

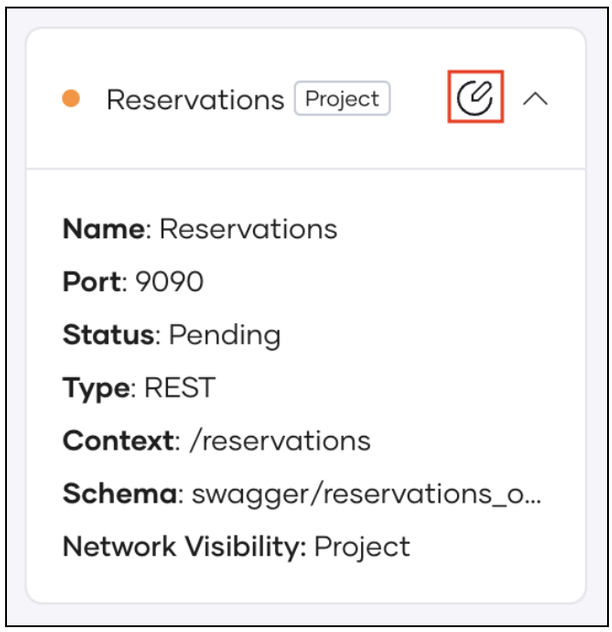
Field Name	Field Value
BuildPack	Ballerina
Ballerina Project Directory	/service

9. Click **Build** on the left navigation menu. Then Click **Build Latest**.

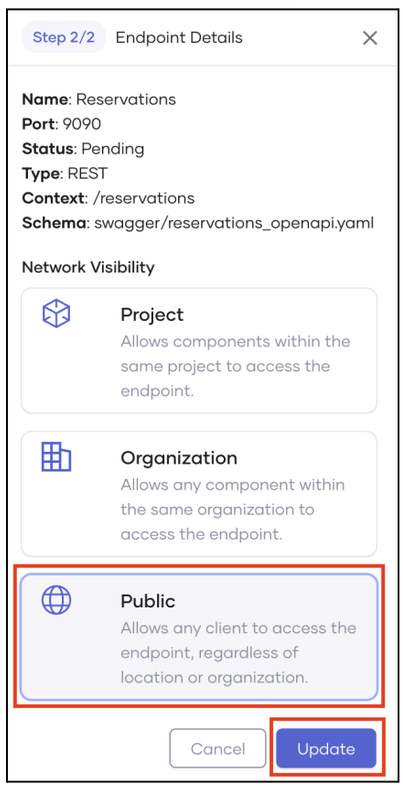
10. Click **Deploy** on the left navigation menu. In the Deploy page, click the **Configure & Deploy**.

11. Click the **Next** button in the right side panel.

12. Click the **Edit** button.



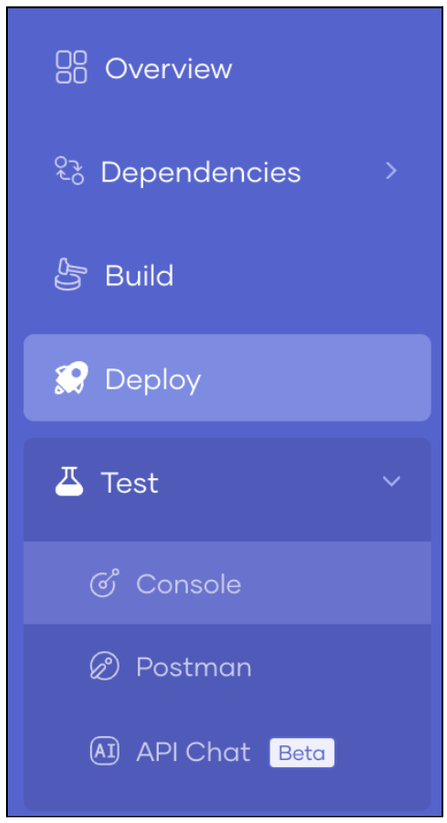
13. Set the Network Visibility as **Public** and click **Update**.



14. Click **Deploy** to deploy the service.

15. Once the service is deployed successfully, click **Promote** to promote the build to the Production environment.

16. Click and expand **Test** on the left navigation menu and click **Console**.



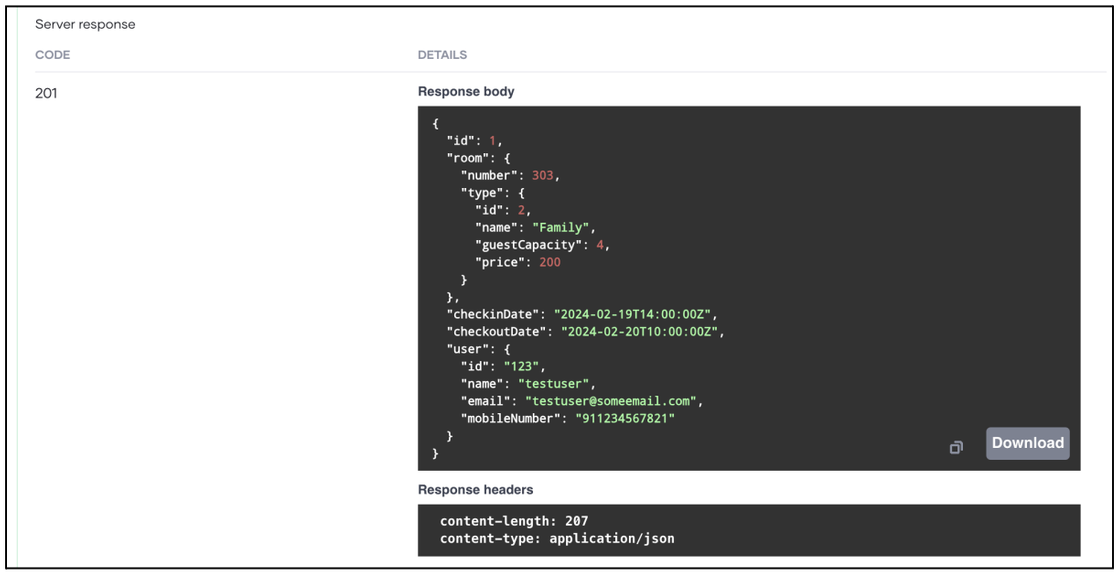
17. Expand the **POST/** section and click **Try it out**. In the **Request body** section, add the following details:

Field Name	Field Value

Request body	{
	"checkinDate": "2024-02-19T14:00:00Z",
	"checkoutDate": "2024-02-20T10:00:00Z",
	"rate": 100,
	"user": {
	"id": "123",
	"name": "testuser",
	"email": "testuser@someemail.com",
	"mobileNumber": "911234567821"
	},
	"roomType": "Family"
	}

18. Click **Execute**.

19. Upon a successful POST request, you will receive a response as shown below.

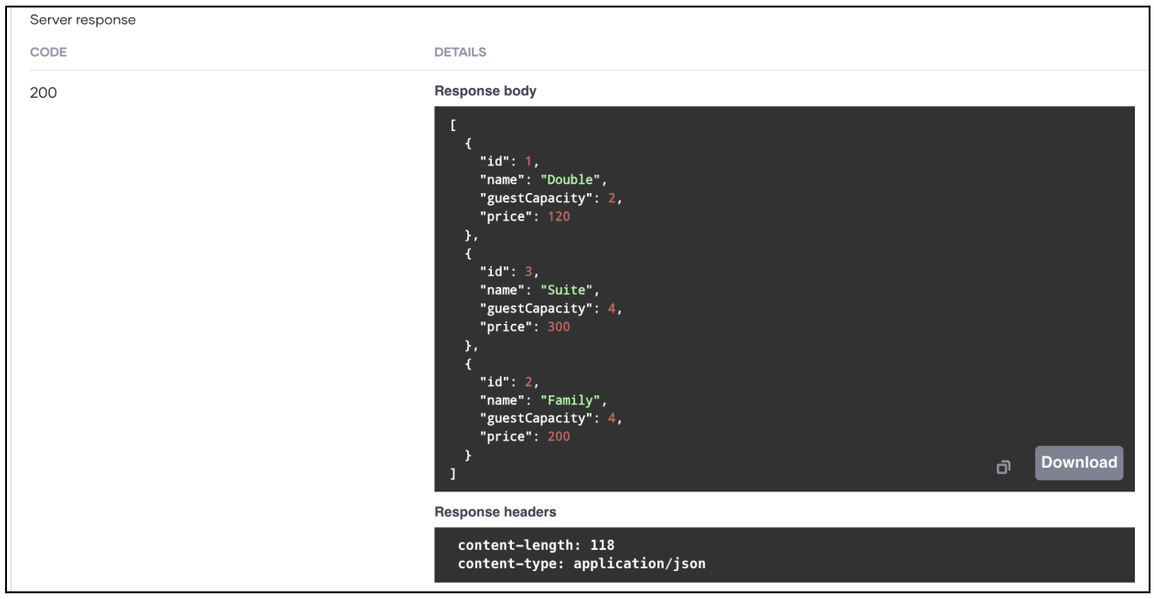


20. Expand the **GET /roomTypes** section and click **Try it out**.

21. Add the following details in each parameter and click **Execute**.

Field Name	Field Value
checkinDate	2024-02-19T14:00:00Z
checkoutDate	
guestCapacity	2

22. Upon a successful GET request, you will receive a response as shown below.



Step 4 - Deploy the Hotel Reservation Web Application

1. From the top main menu, select the **Project** tab.

2. Click **Create**.

3. Select the **Web Application** card.

4. Provide a name for the Web Application.

Field Name	Field Value
Name	Hotel Reservation Web Application

5. Click on the **Authorize With GitHub**.

6. Select the **hotel-reservation-demo** repository and select the branch **complete**.

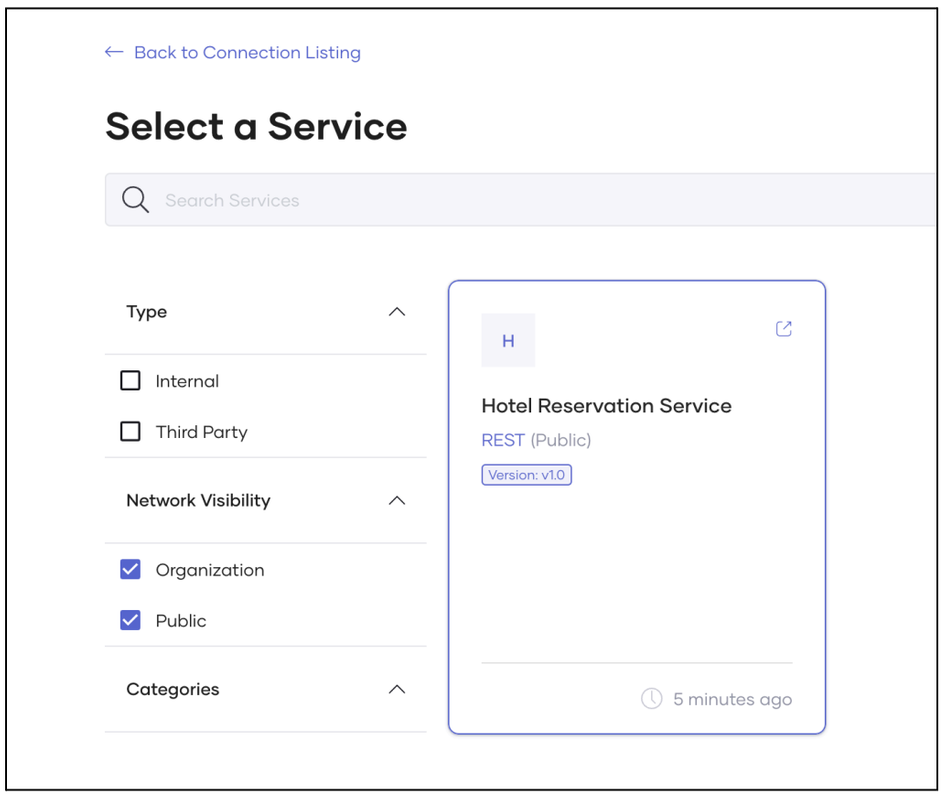
7. Select **React** as the **Buildpack** and set the following details. Then click **Create**.

Field Name	Field Value
BuildPack	React
Project Directory	/webapp
Build Command	npm run build
Build Path	/build
Node Version	20.11.0

8. Click and expand **Dependencies** on the left navigation menu and click the **Connections** tab.

9. Click **Create**.

10. Select the **Hotel Reservation Service** that was created in step 3.

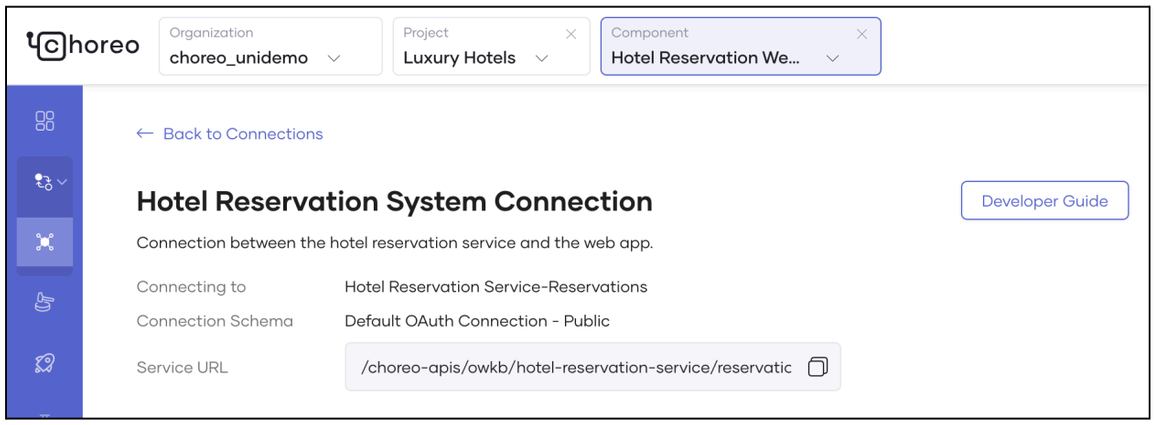


11. Provide a name and a description for the connection.

Field Name	Field Value
Name	Hotel Reservation System Connection

12. Then click **Create**.

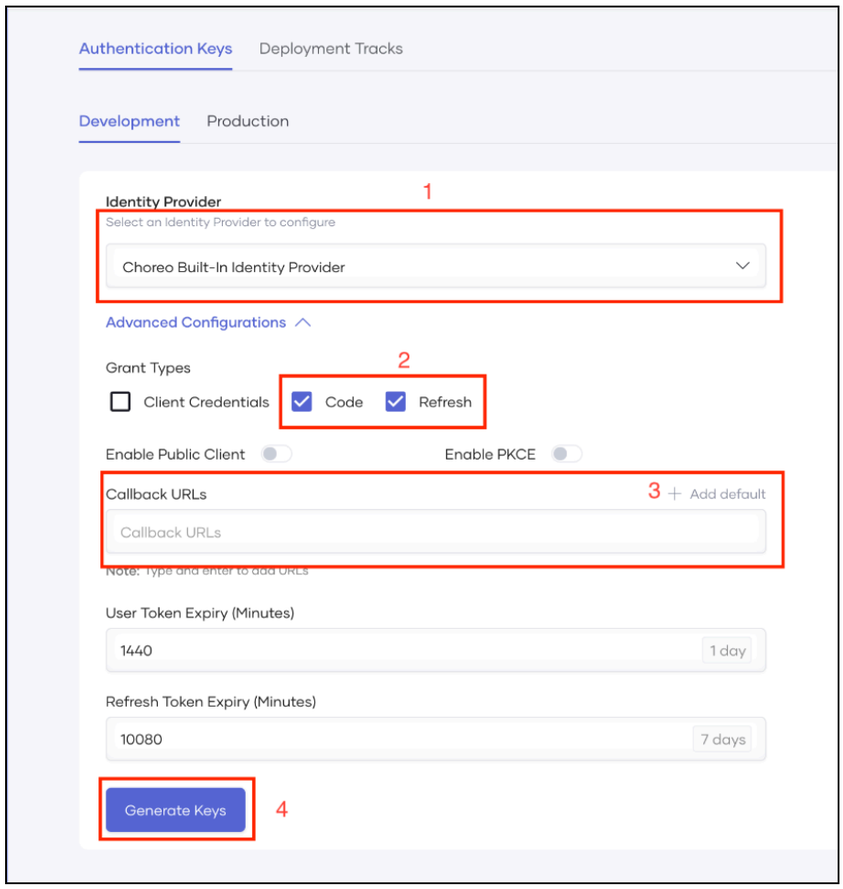
13. Copy and save the **Service URL** for later use.



14. Click **Build** on the left navigation menu. Click the **Build Latest**.

15. Once the build is completed, Click **Settings** on the left navigation menu.

16. Under the **Development** tab, select Choreo Built-In Identity Provider as the Identity Provider. Expand the **Advanced Configuration** section. Select the **Code** and **Refresh** grant types. Click **Add Default** to add a default callback url. Then click **Generate Keys**.



17. Click **Deploy** on the left navigation menu. Click the **Configure & Deploy** in the Deploy page.

In the right side panel, provide config.json as the Configuration File Name and provide the following content:

```
Unset
   window.configs = {
           apiUrl: '<Service URL>',
      };
```

Replace <Service URL> with the value that you copied when creating a connection to the Service in **Step 13**.

Ex:

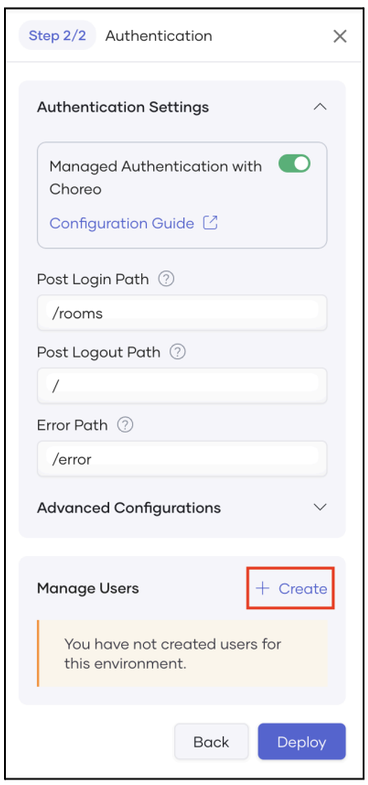
```
Unset
window.configs = {
      apiUrl:
'/choreo-apis/owkb/hotel-reservation-service/reservations-365/v1.0',
};
```

18. Click **Next.**

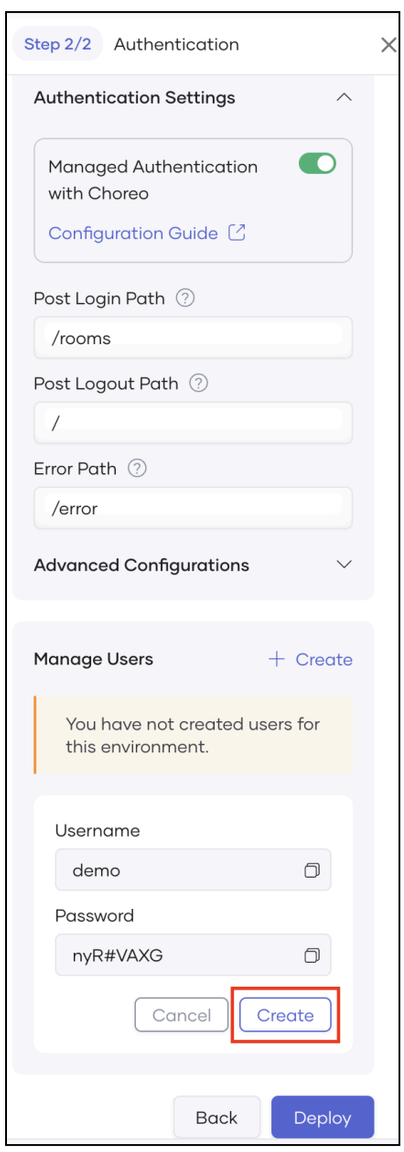
19. Provide following values for each parameter in the Authentication Settings window.

Field Name	Field Value
Post Login Path	/rooms
Post Logout Path	/
Error Path	/error

20. Click **Create** to expand the demo user creation window.



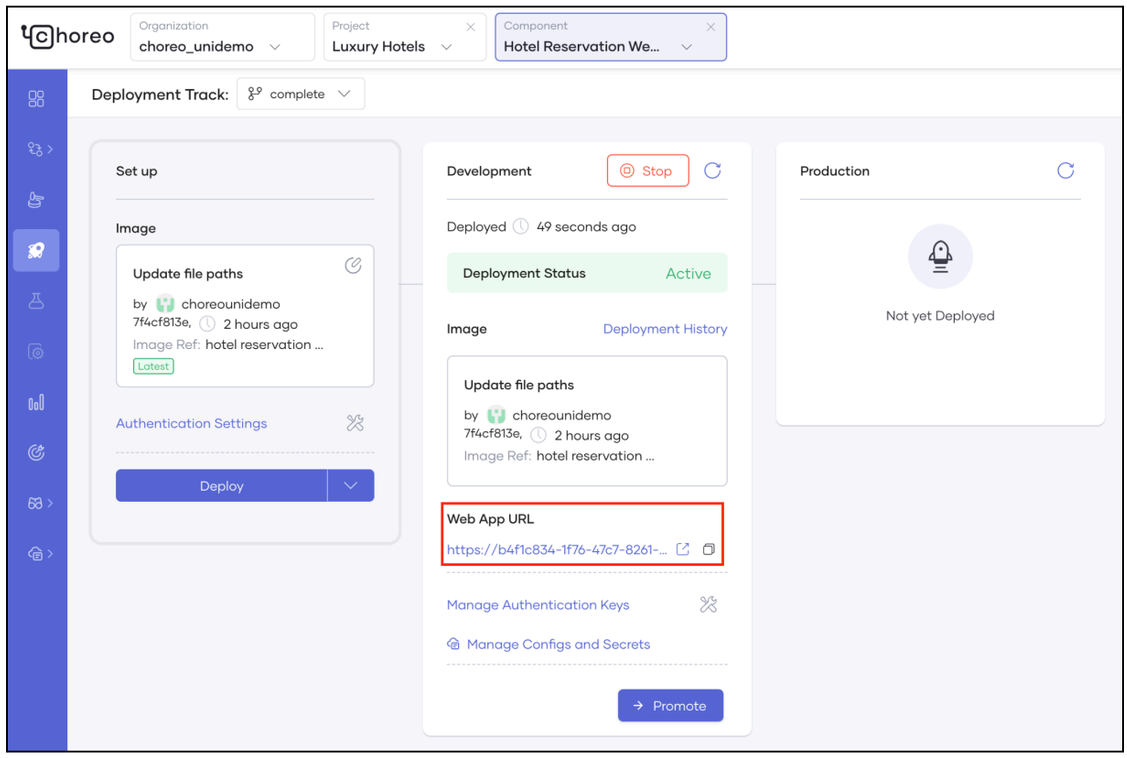
21. Click **Create** to create the user.



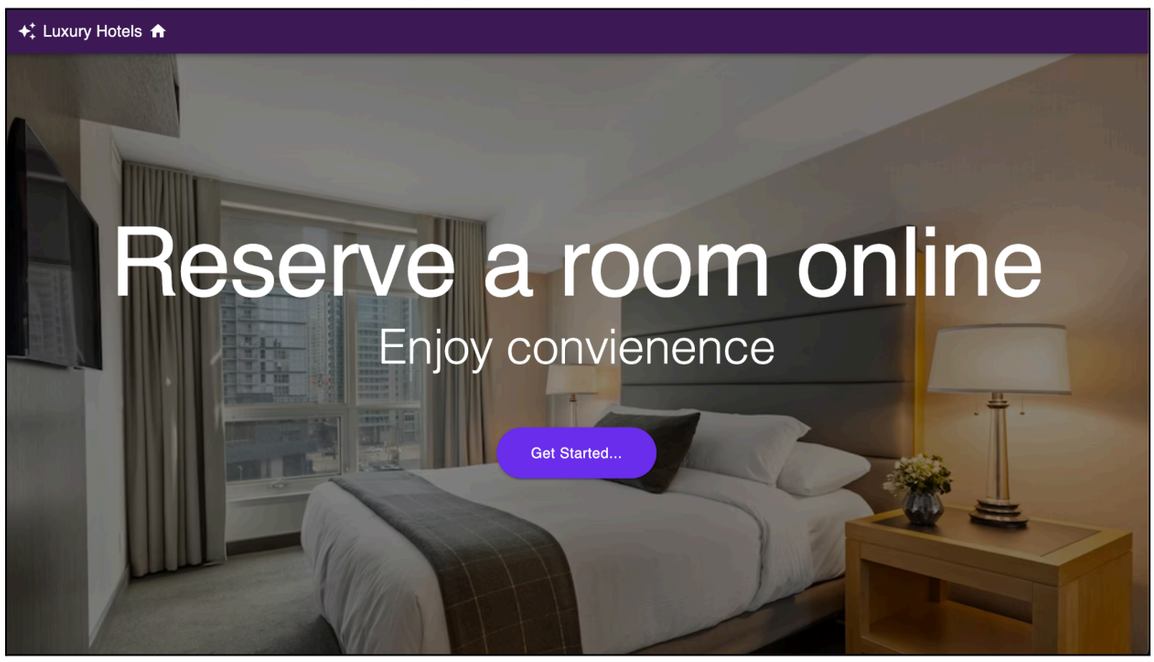
22. Once the user is created, save the username and the password for later use.

23. Click **Deploy** in the right side panel to deploy the web application.

24. Once the web application is deployed successfully, click on the **Web App URL** to access the web app.

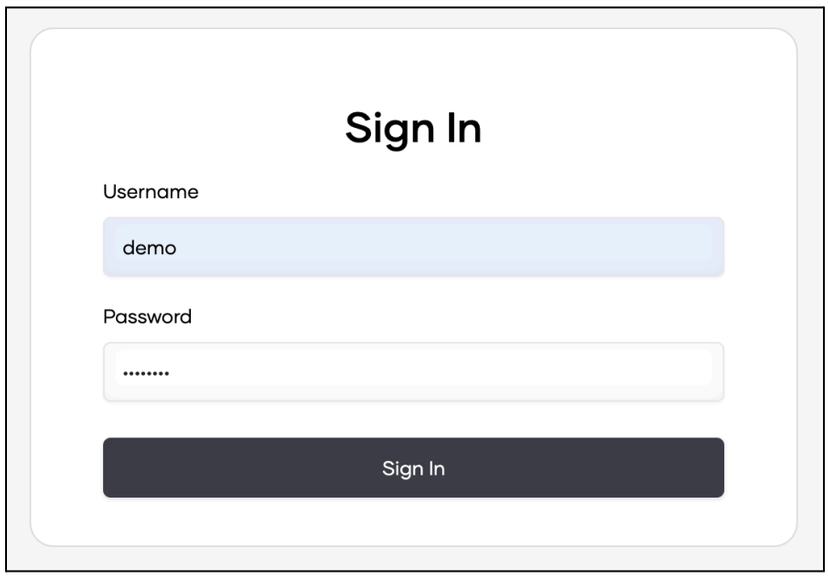


25. You will be greeted with the landing page shown below:

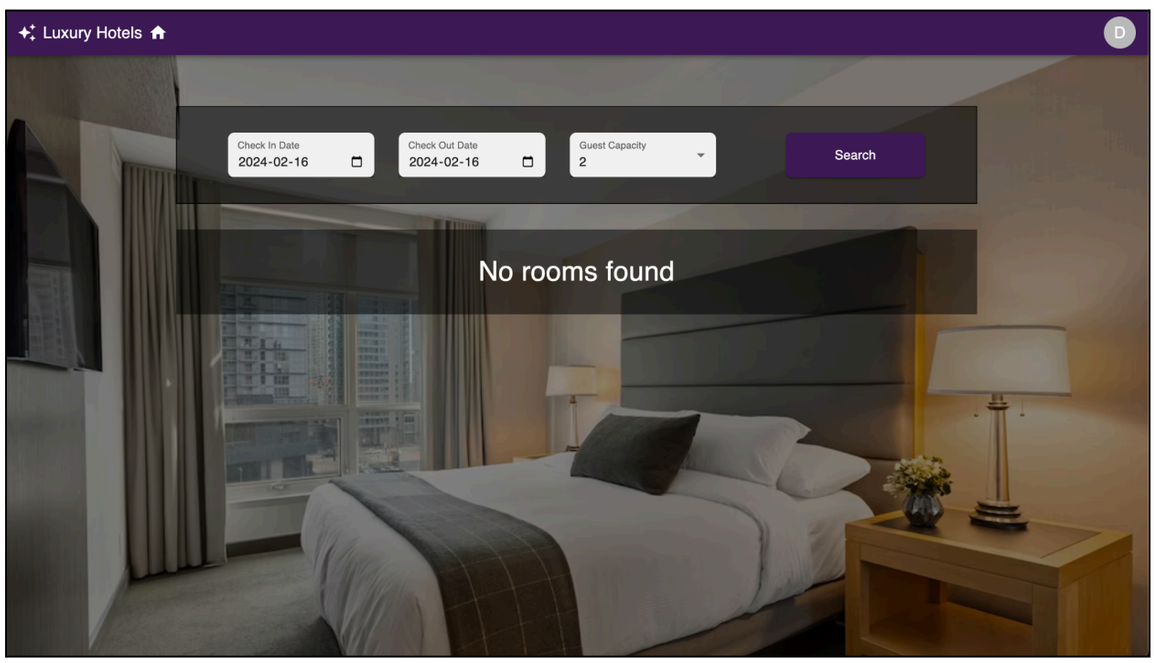


26. Click the **Get Started...** button.

27. This will direct you to the login page. Provide the demo username and password (created in the Step 21) and click Sign In.

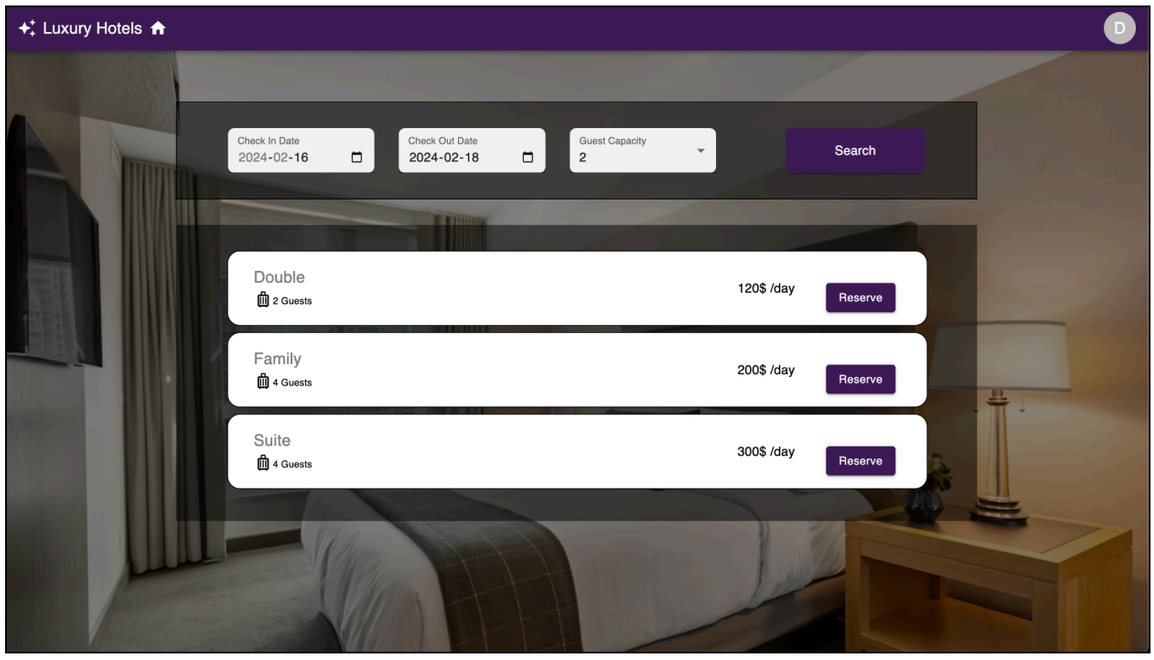


28. You will be logged into the Hotel Reservation web application.



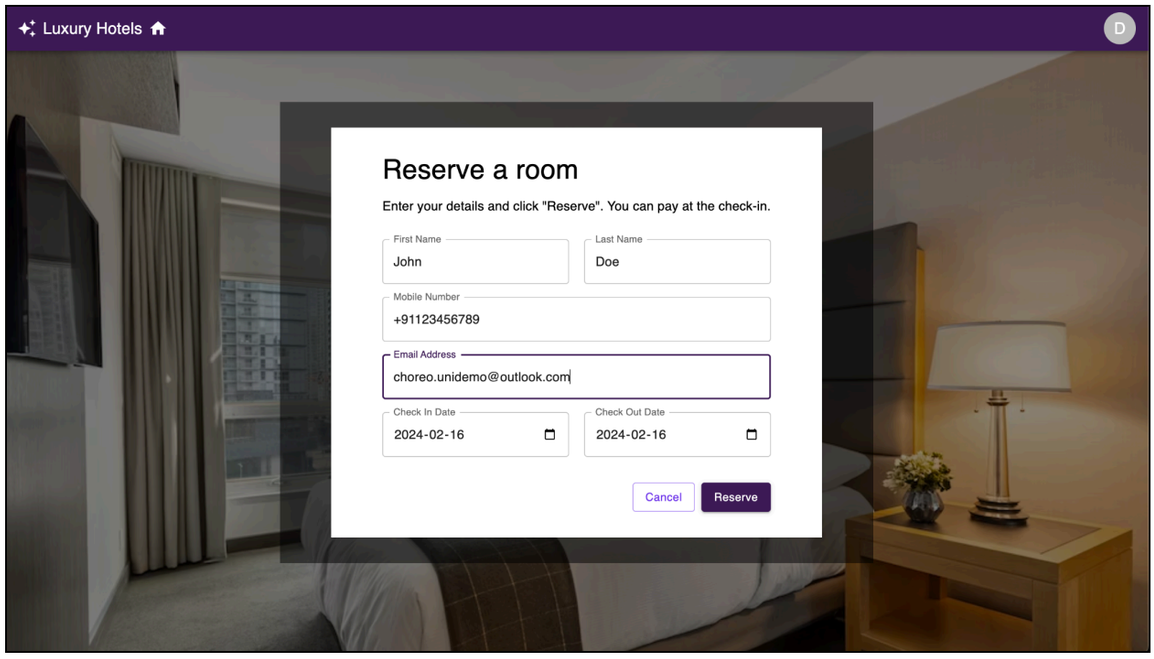
29. Set a check in date, check out date and select the no of guests. Then click **Search**.

30. You will be able to get the available rooms for the selected date range and guest capacity.



31. Click **Reserve**.

32. Provide required details as below and click Reserve.



33. Your reservation will be created and you will receive an email confirmation.