

## Project

### **Description:**

A restaurant chain has reached out to your team to build a reservation system.

### **Here are the details:**

- Two categories of users / customers: guest user or registered user.
- Users should be able to search for a table and reserve.
  - o User doesn't need to login to the system to reserve a table. If registered users, they can login.
  - o User enters name, phone, email, date and time (date picker), and # of guests for dining and system presents available tables.
  - o Tables have maximum capacity limit i.e., 2, 4, 6, or 8.
  - o Different combinations are allowed, and owner accommodates the seating, for example: someone requests 8 guests and table for 8 is not available but 2 + 6, or 4+4 is available. System should combine the tables and notify owner they need to combine tables. In this case System reserves both tables.
- If a guest user i.e., not a registered user, system should prompt user to register (Optional) before finalizing the reservation.
- Registered users will have these fields:
  - o Name, mailing address, billing address (checkbox if same as mailing address), Preferred Diner # (system generated), Earned points (based on \$ spent i.e., \$1 is 1 point), preferred payment method (cash, credit, check).
- System should track high traffic days / weekends and a hold fee is required i.e. July 4<sup>th</sup> will require valid credit card on system to reserve the table.
  - o Notify user no show will have minimum \$10 charge.

### **Assumptions:**

If you make any assumptions to provide good user experience, please list it.

**Assuming the restaurant layout for tables**

### **Answer these questions:**

1. Implement the web or mobile application following the SE best practices we covered in this class.
2. You should have already picked the tools and technologies and provided the details in assignment 3. If you are changing it, provide reasoning. Whatever

programming language / framework you pick, make sure you are familiar with it. When you are stuck, the group should find the solution.

Our team mentioned that we would be utilizing Python as part of our techstack, but there was really no purpose of using Python since we were able to use javascript for both the frontend and backend.

3. Working and demonstrable code.

## LINK TO OUR GITHUB REPOSITORY

<https://github.com/yanelya/software-engineering-project>

---

### Reservation (No table combination)

```
_id: ObjectId('638a8f3ac31bad3b8c752a81')
first: "x"
last: "x"
phone: "123-456-7953"
email: "x@x.com"
date: "Mon Dec 05 2022"
time: "12:00 PM"
guests: "1"

```

This is an object in our database that demonstrates a reservation entry made in the frontend. You can see how a user who made a reservation with 1 guest was assigned to a table 1 that supports 2 guests. No table combinations were needed since there were no reservations that conflicted with users' guests number.

## Combined Tables Example

```
_id: ObjectId('636f3c18ad00c314b7f42dea')
first: "Yanely"
last: "Ayala"
phone: "832-875-6867"
email: "yanelylayala@gmail.com"
date: "Fri Dec 02 2022"
time: "11:00 AM"
guests: 2
table_number: 3
```

```
_id: ObjectId('636f3ddbbc51a15b301520fa')
first: "Nate"
last: "King"
phone: "567-889-6834"
email: "NateKing@gmail.com"
date: "Fri Dec 02 2022"
time: "11:00 AM"
guests: 5
table_number: 4
```

These are 2 reservations in the database that are for the same date and time. If a user were to make a reservation with the same details, the only available tables would be tables 1 & 2 (where table 1 supports 2 people, and table 2 supports 4 people).

```
_id: ObjectId('638a8ef8c31bad3b8c752a7d')
first: "jake"
last: "fromstatefarm"
phone: "123-456-7890"
email: "jakefromstatefarm@gmail.com"
date: "Fri Dec 02 2022"
time: "11:30 AM"
guests: "5"
|  |  |
| --- | --- |
| table_number: | Array |
| 0: | Object |
| _id: | "637ef0fd0f7cc64126f4bf20" |
| seats: | 6 |
| table_number: | 3 |


__v: 0
```

This reservation shows a combination made since a user wanted to make a reservation for 5 guests when only tables supporting 2 & 4 people were available. Since the summation of the available tables were greater than or equal to the number of guests asked, we were able to create a reservation with the given combination of tables.

**Here is a demonstration of our tables combination reservation form and our notifications to register and no-show warning for busy days**

<https://youtu.be/jTocm5NutmI>

## Register Form

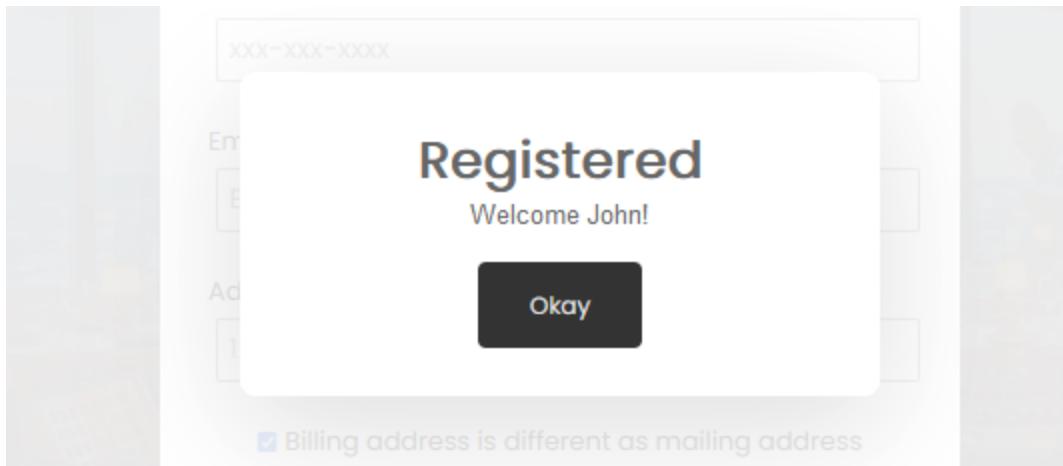
Registering without billing address:

The screenshot shows a mobile application interface. On the left, a "Register" screen displays input fields for First Name (John), Last Name (Smith), Phone Number (182-848-8348), Email (johnsmith@legitamteemail.com), and Address (john smith road 1234). Below these fields are two checkboxes: "Billing address is different as mailing address" (unchecked) and "Choose preferred payment method" (set to "Cash"). A "Submit" button is at the bottom. On the right, a semi-transparent "Confirm to submit" dialog box is overlaid. It contains the user's information: First Name: John, Last Name: Smith, Phone Number: 182-848-8348, Email: johnsmith@legitamteemail.com, Mailing Address: john smith road 1234, and Payment Method: Cash. At the bottom of the dialog are "Yes" and "No" buttons.

Registering with billing address:

This screenshot is similar to the previous one but includes a checked checkbox under "Billing address is different as mailing address". The "Billing Address" field now contains "993-499-3943". The rest of the registration fields and the confirmation dialog are identical to the first screenshot.

Click yes and we are now registered

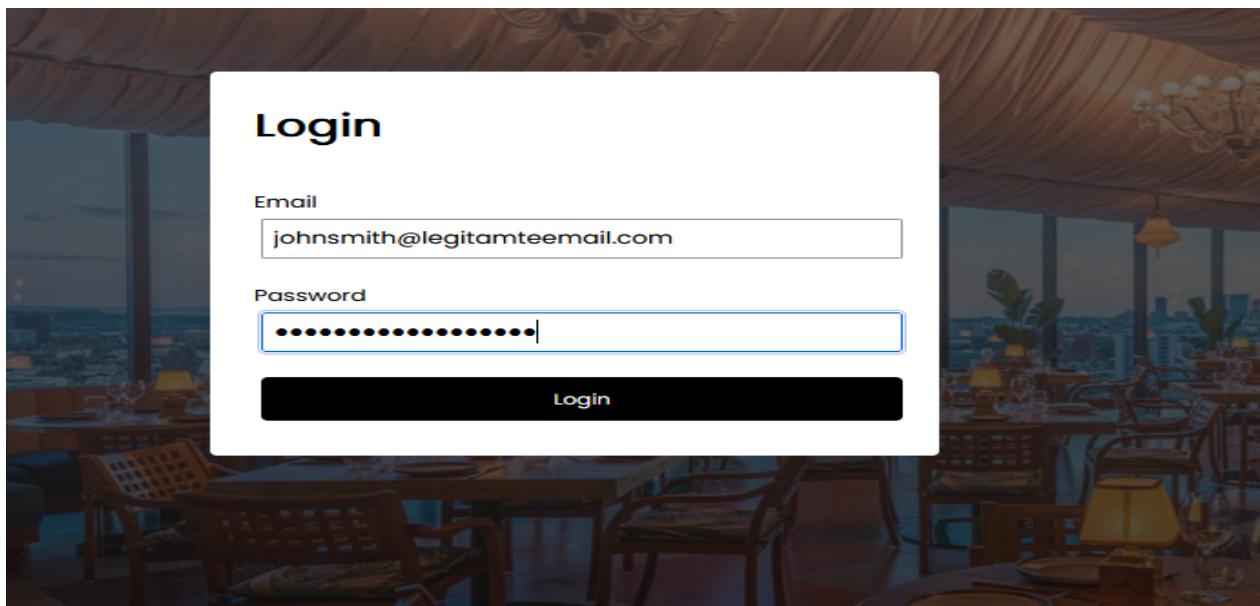


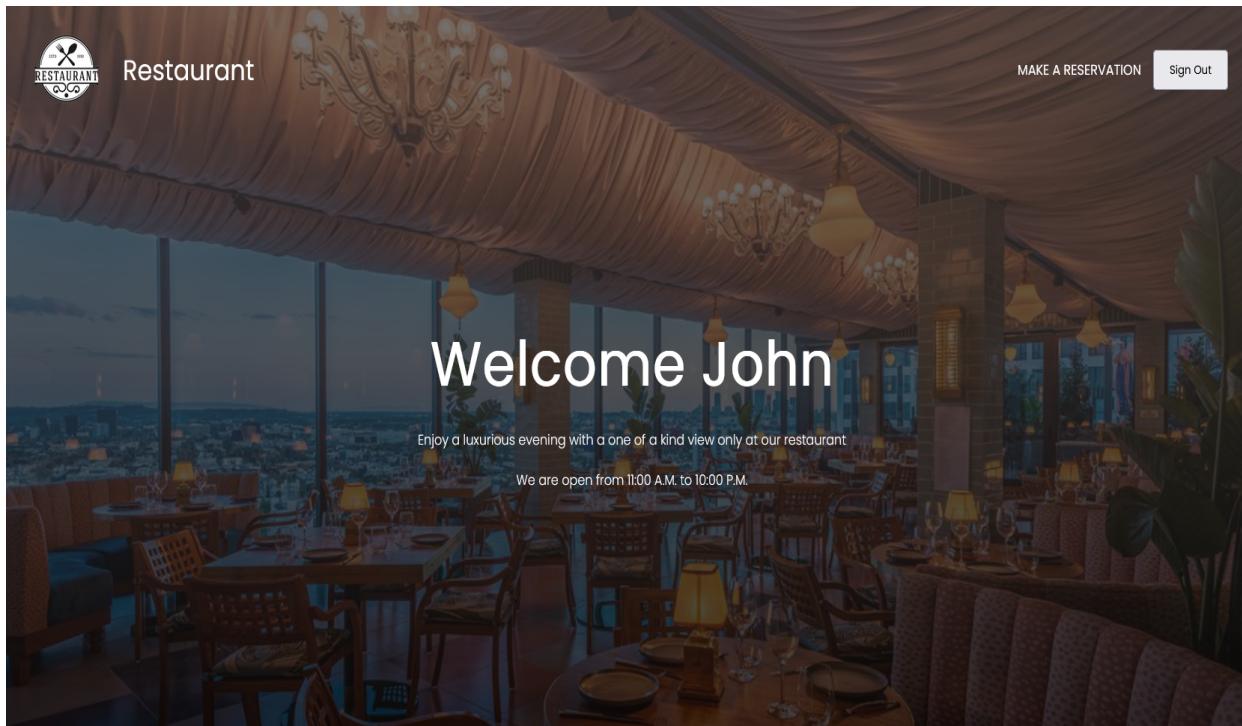
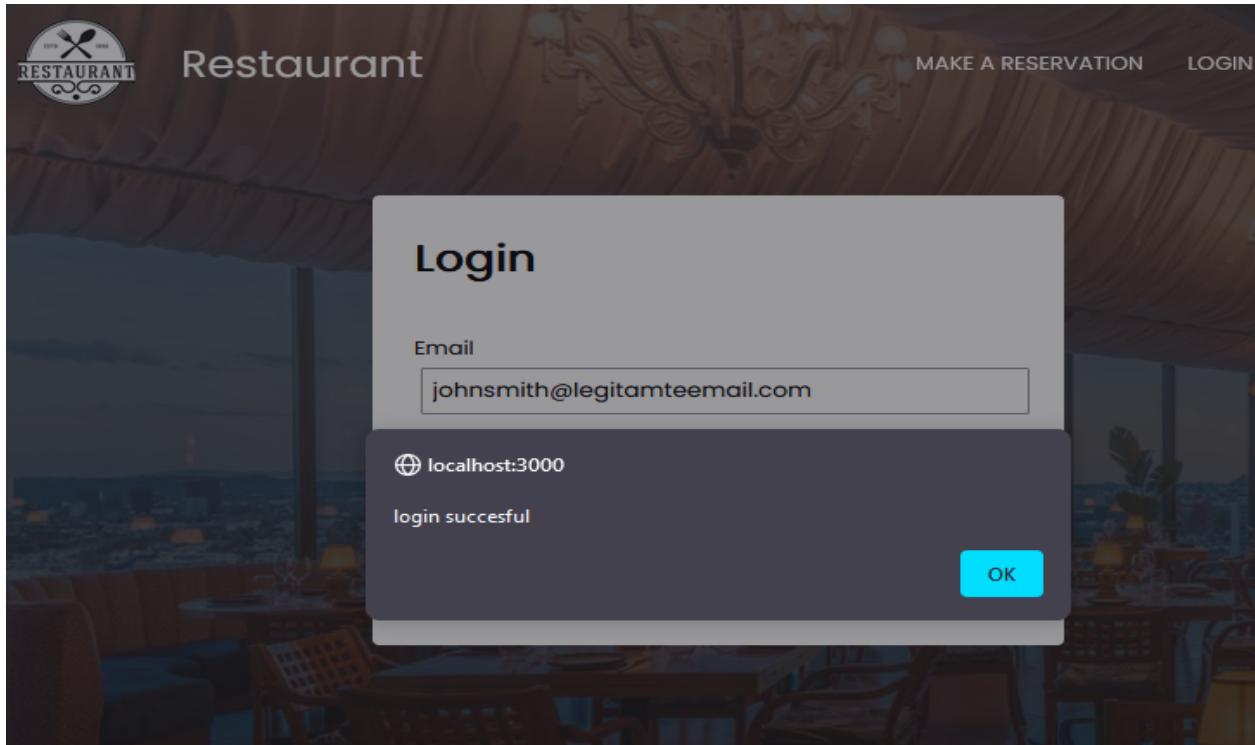
```
_id: ObjectId('638a9b2a1b6816884d1b96ce')
first: "John"
last: "Smith"
phone: "182-848-8348"
email: "johnsmith@legitamteemail.com"
billing_address: "993-499-3943"
mailing_address: "john smith road 1234"
password: "Johnsmithpassword1"
payment_method: "Cash"
points: 0
preferred_diner: 0
__v: 0
```

As you can see john smith was now correctly added to the database

## Login Form

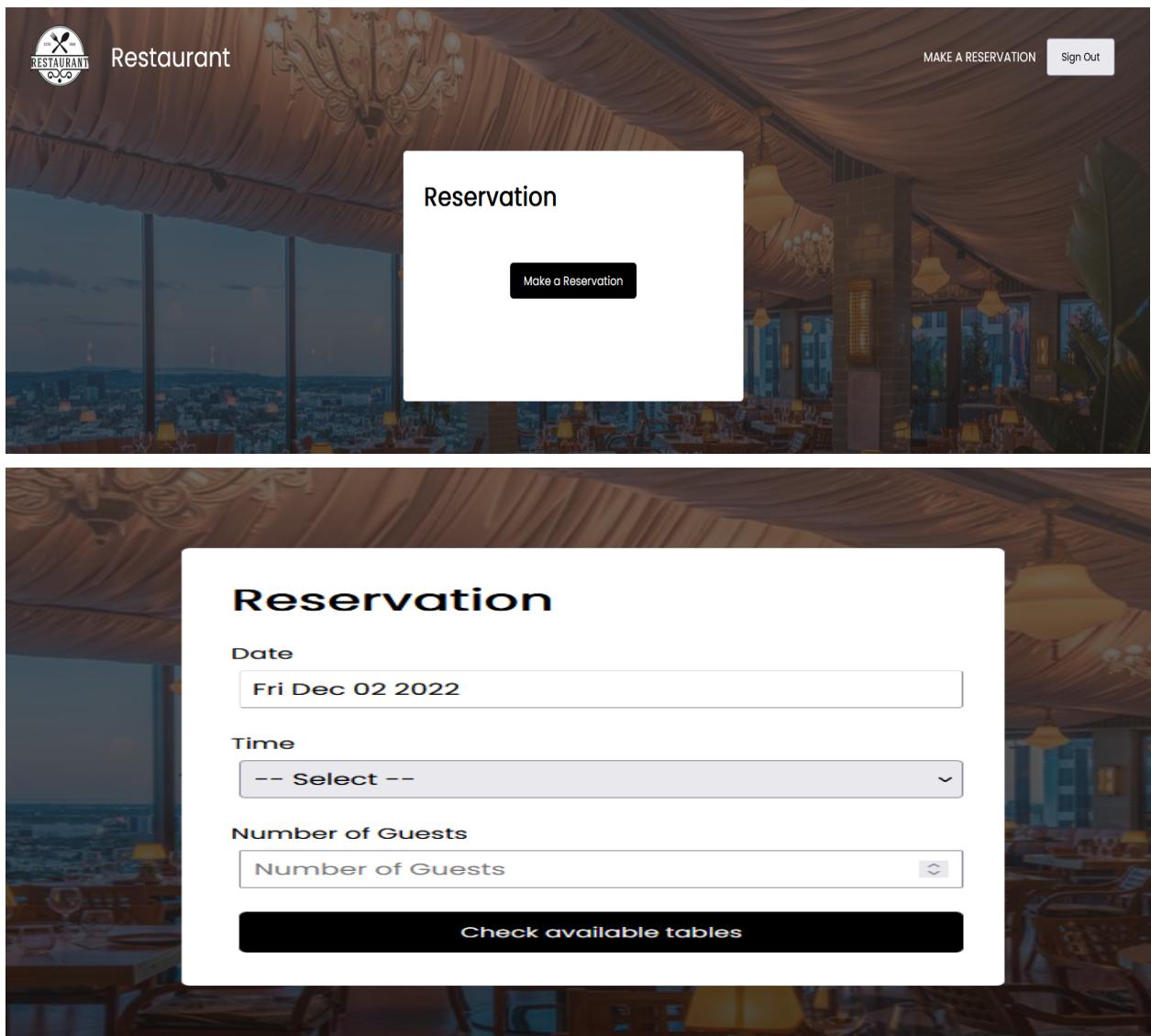
Now lets login as john





Now let's make a reservation

## Reservation Form (With Login Information)

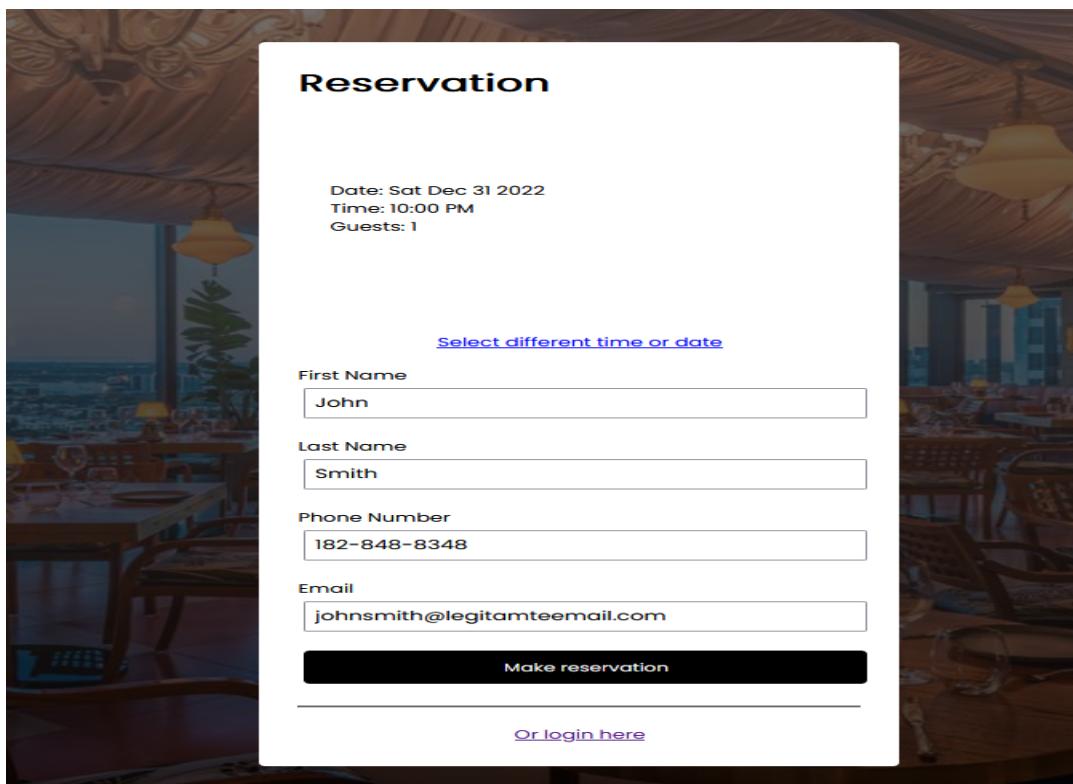
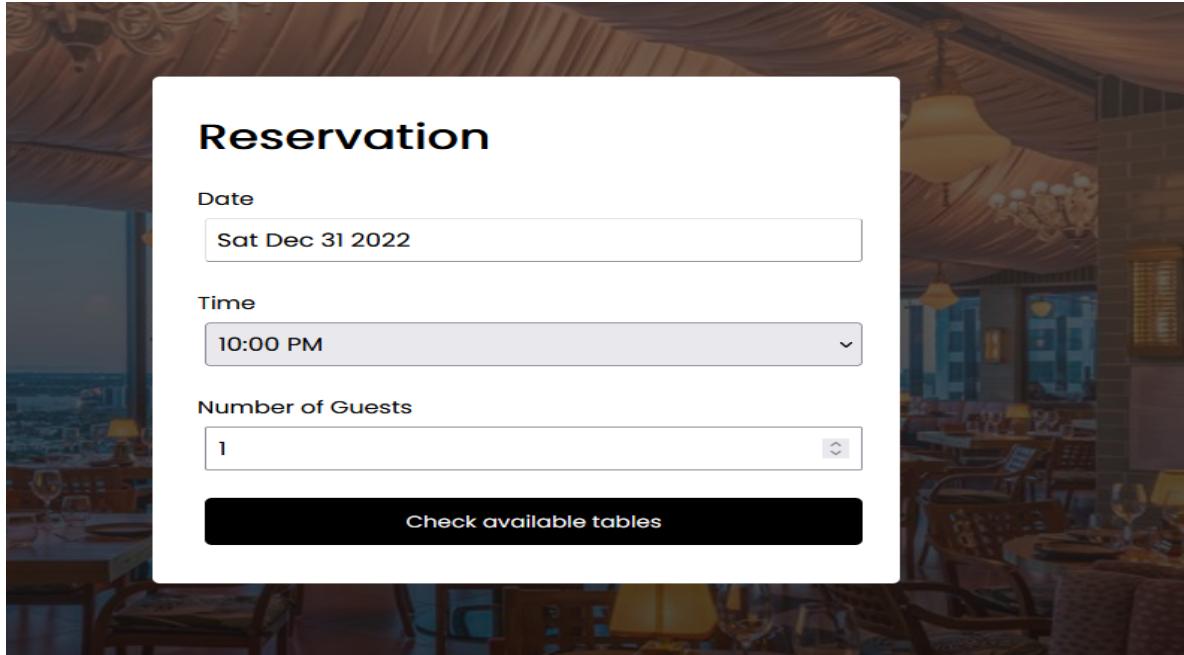


The image shows a restaurant's interior with large windows overlooking a city skyline at dusk. A white reservation overlay is centered in the foreground. The overlay has a dark border and contains the word "Reservation" in bold black font at the top. Below it is a dark button with the text "Make a Reservation".

The reservation form is shown in two states:

- Initial State:** Shows the title "Reservation" and a single button "Make a Reservation".
- Filled State:** Shows the title "Reservation" and three input fields:
  - Date:** A text input field containing "Fri Dec 02 2022".
  - Time:** A dropdown menu showing "-- Select --".
  - Number of Guests:** A dropdown menu showing "Number of Guests".A large black button at the bottom of the form says "Check available tables".

Let's choose a day, time, and specify the amount of guests



We can now make reservation with autofilled user information

## Reservation

Date: Sat Dec 31 2022  
Time: 10:00 PM  
Guests: 1

### Confirm reservation

Last Name

Smith

Phone Number

182-848-8348

Email

Click yes

## Reservation

Date: Sat Dec 31 2022  
Time: 10:00 PM  
Guests: 1

### Reservation created!

No show will have minimum \$10 charge

Last Name

Smith

Phone Number

182-848-8348

Email

johnsmith@legitamteemail.com

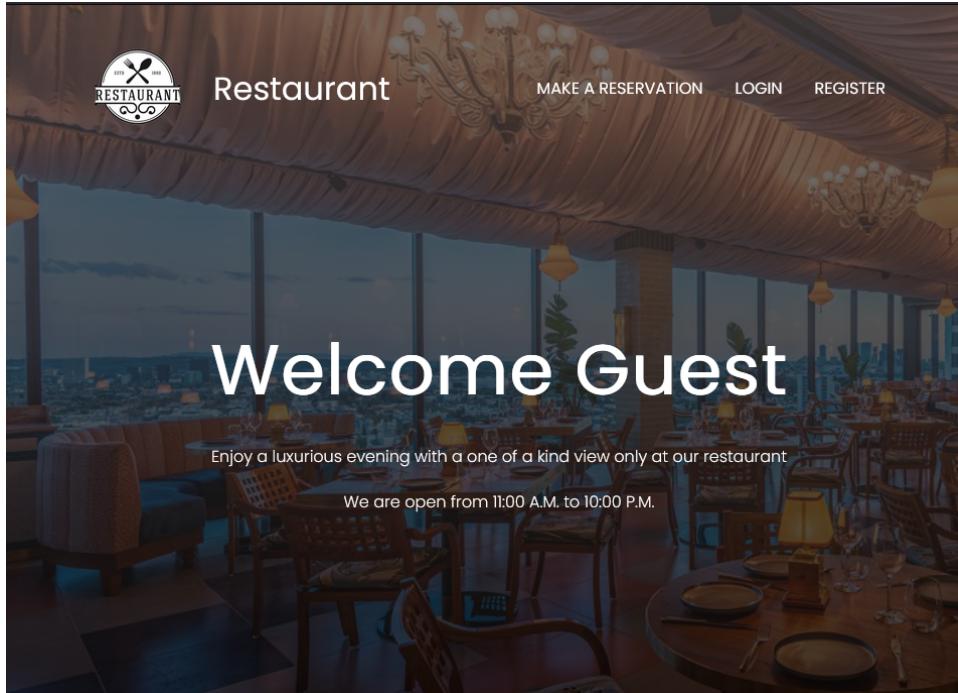
[Or login here](#)

Since our reservation is a holiday “no show” notification comes up.

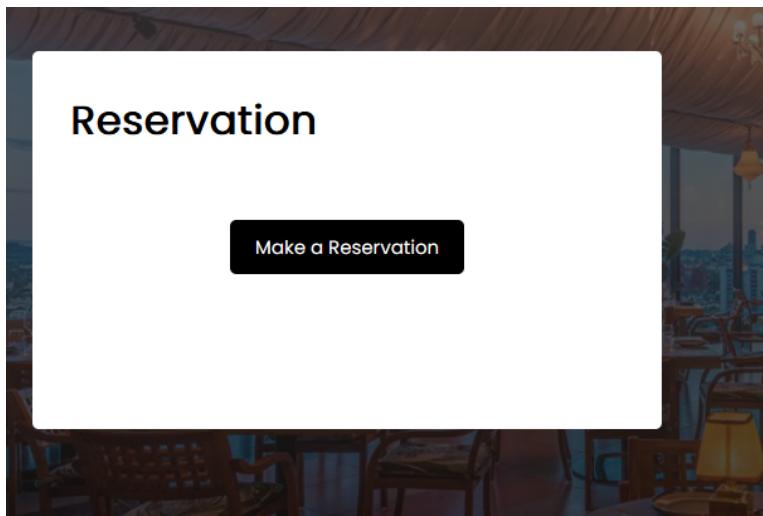
```
_id: ObjectId('638a9f931b6816884d1b9710')
first: "John"
last: "Smith"
phone: "182-848-8348"
email: "johnsmith@legitamteemail.com"
date: "Sat Dec 31 2022"
time: "10:00 PM"
guests: "1"
> table_number: Array
__v: 0
```

There is now a john smith reservation in the database at the time specified.

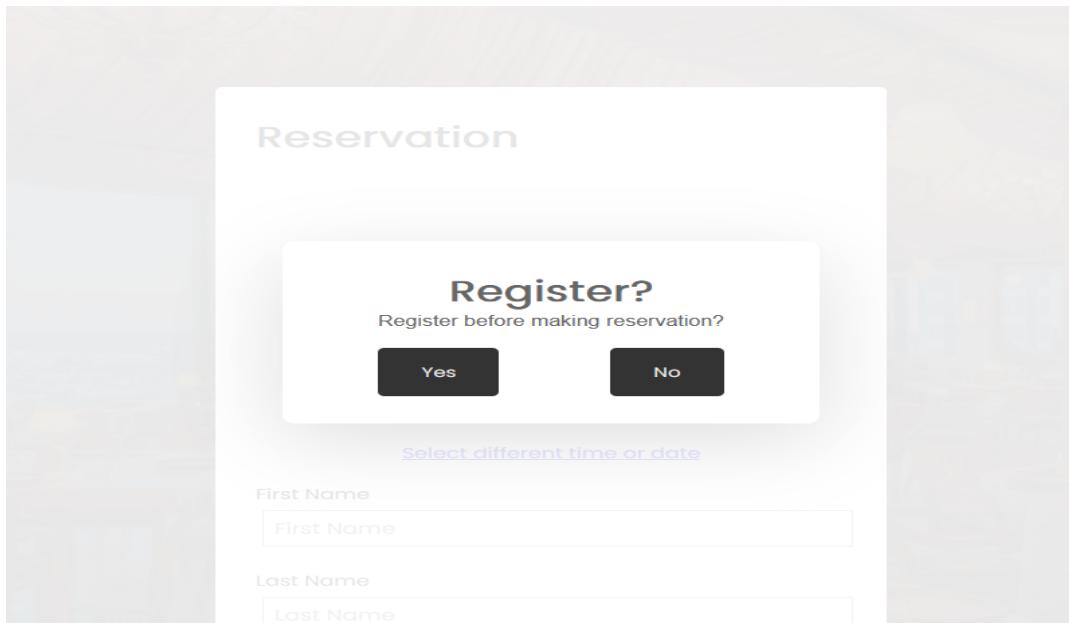
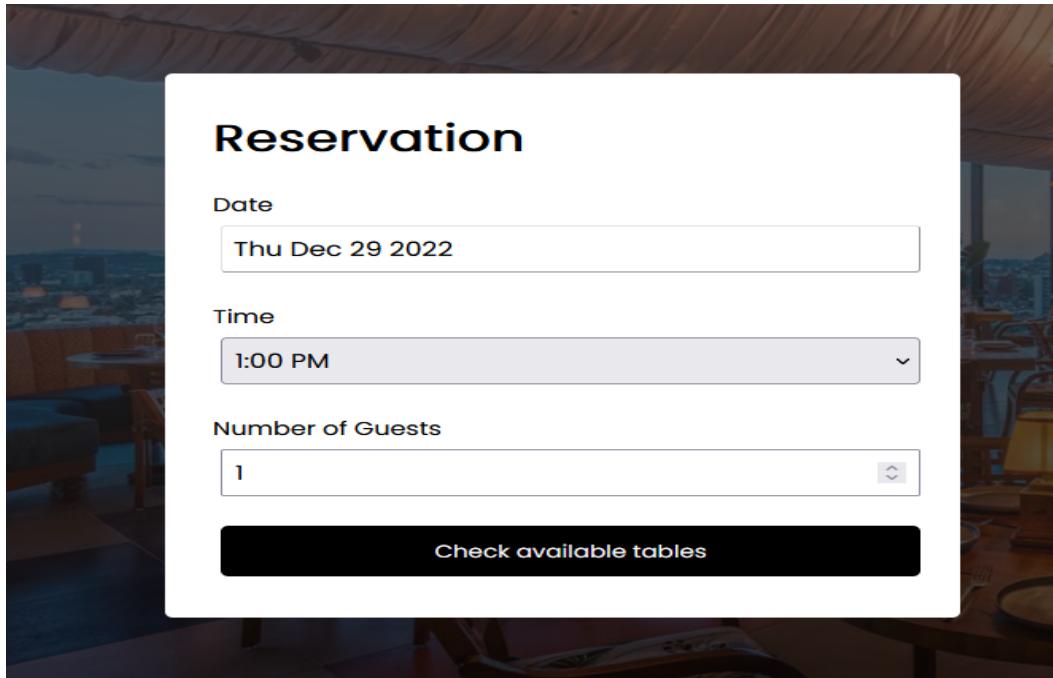
### Reservation Form (Without Logging in)



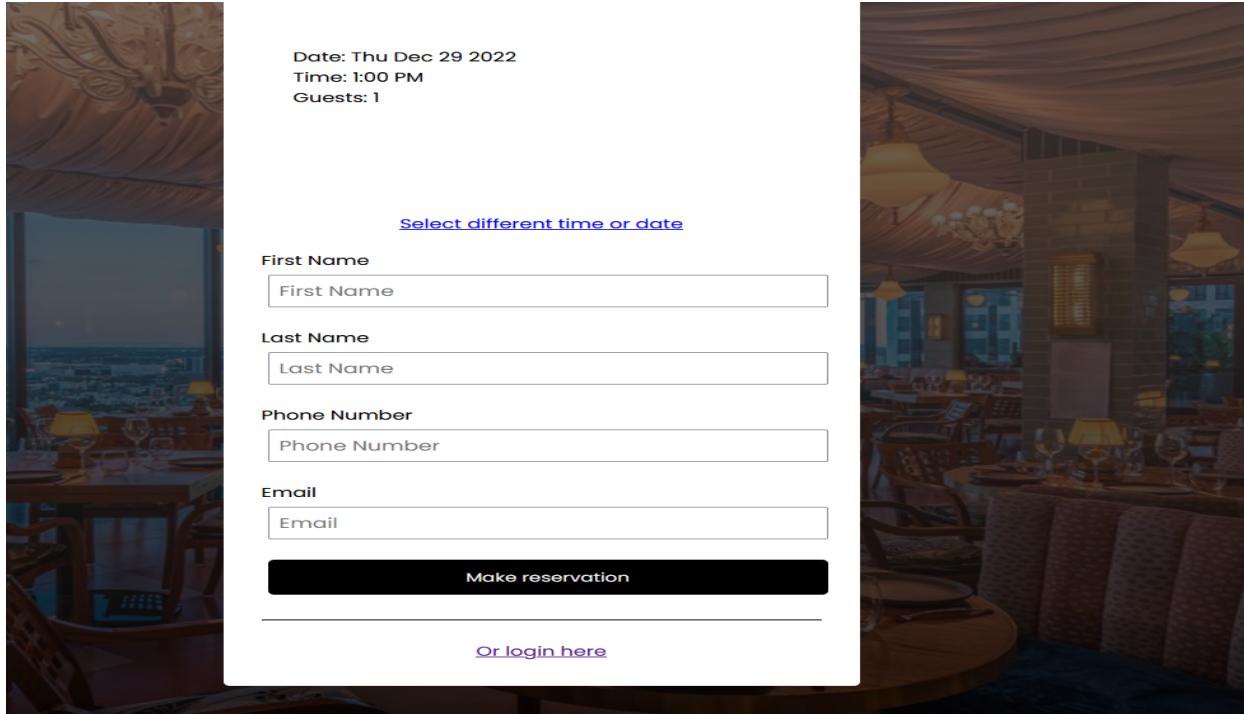
If you are not logged in you are greeted as a guest. Let's make a reservation



Let's specify the date, time and amount of guests for our reservation



It now asks us to register before making the reservation.



Since we are guests there is no information to autofill. The rest of the reservation process is the same as when logged in.

## Unit Tests

```
1  test('canary test', () => {
2      expect(true).toBe(true);
3  });
4
```

```
import React from "react";
import ReactDOM from "react-dom";
import Login from "../components/LoginForm";

it("renders without crashing", ()=>{
  const div = document.createElement("div");
  ReactDOM.render(<Login>/<Login>, div)
})
```

Running npm tests in the frontend directory gives us:

```

PASS  src/_tests__/Canary.test.js

Test Suites: 2 passed, 2 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        2.509 s, estimated 3 s
Ran all test suites.
PS D:\Joaquin de Leon\This HDD\Documents\GitHub\software-engineering-project\frontend>

```

4. Demo with TA. Each team member must participate.

**REQUIRED:**

Fill in this table, provide as many details as possible:

Group Member Name	What is your contribution?	Discussion Notes
1. Joaquin de Leon	Worked on all Log-in logic and helped with checking what days were busy (weekends and holidays)	
2. Yanely Ayala	Set up backend & frontend. Created reservation system including table reservations and table combination system.	
3. Yusra Shaikh	Worked on forms and form validations for input fields, frontend design, and tests	

**What to turn in:**

- Only soft copies of UI screenshots uploaded to blackboard before due date.
- No extensions.
- Frequently check in to GitHub.
- Each team member must equally contribute.
- To get full credit provide details and diagrams (**when appropriate**).

**Notes:**

You can start early on this project along with assignments.