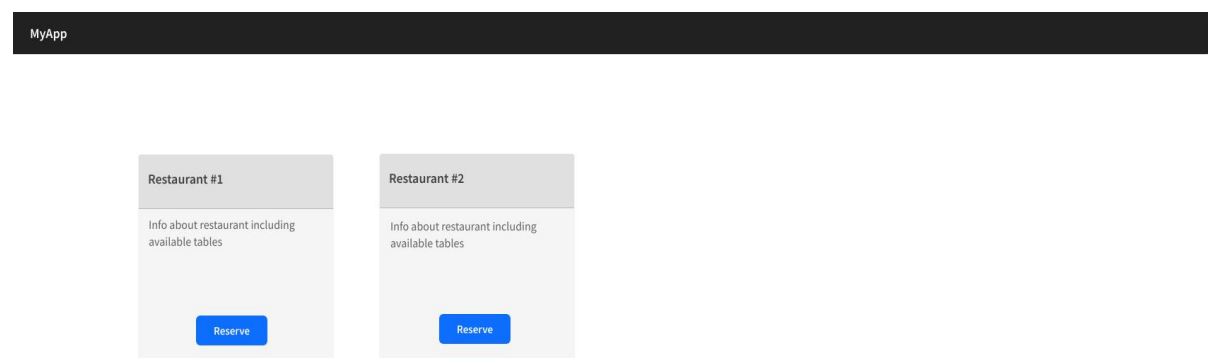


Muhammad Ali Shahzaib
2020463

In today's world, dining in hotels and restaurants is becoming more and more difficult as they are running out of spaces all the time. Upon pondering on it for a while, I came up with the idea to develop an application for booking tables in different restaurants as quickly as possible.

I took my time to research what do common seat reservation applications offer and took notes of the most efficient features as I wanted my design to be very sleek. I took the concept of displaying all hotels from **FoodPanda**, table reservation idea from **Chai Cafe**, and the 'All reservations' view idea from **CPanel**.

Lets start off with the design. After going over several designs, I decided that I want the user to be greeted by all the available restaurants in our database that offer bookings. For this, the section on the page with restaurants is dynamically populated depending on the number of records in our database. It offers a very simple and uniform design with very less clutter.



You can see from the wireframe that I was not decided on the background until the very last stage of development, only because I wanted the screen to look simple.

Upon selecting the restaurant, the user can reserve a table by clicking the appropriate button. This brings the user to a new page where he needs to fill some information that will be saved for this reservation. It is a form consisting of fields that require info such as your name, date of birth, contact and email etc. You cannot leave any field blank.

The wireframe shows a form titled 'Enter Details'. It contains several input fields with labels and validation feedback:

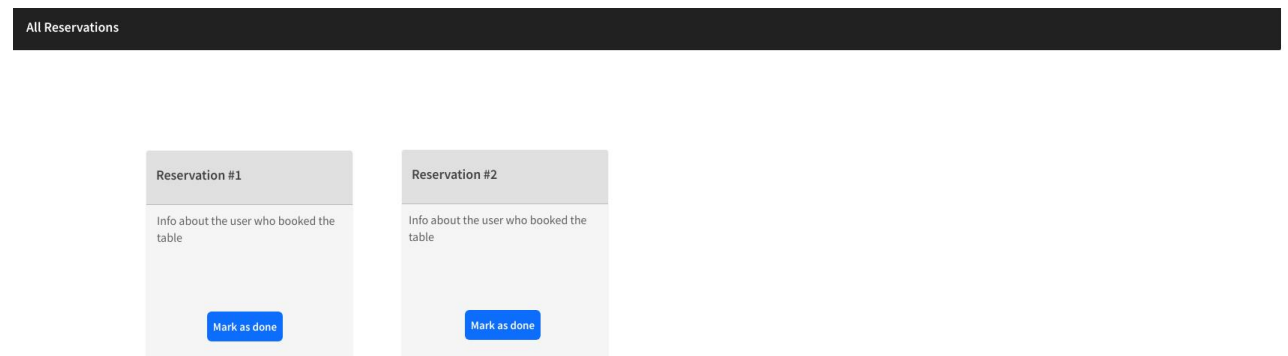
- First name:** Input field with 'Mark' and a green checkmark. Feedback: 'Looks good!'.
- Last name:** Input field with 'Otto' and a green checkmark. Feedback: 'Looks good!'.
- Email:** Input field with '@' and a red outline. Feedback: 'Please choose a username'.
- Address:** Input field with a red outline. Feedback: 'Please provide a valid city'.
- Hotel Name:** Label with a 'Tag line' below it.

At the bottom center is a blue 'Submit' button.

Upon clicking the submit button, the input data is stored in local JSON files, but this can be integrated with cloud based databases such as MongoDB and Firebase so that it is accessible from anywhere. If the data is stored successfully, the user gets an alert on screen that the table has been booked.

As soon as the user closes this window, the count of available tables for that restaurant on the main screen decreases by 1.

To view all the reservations that are currently booked, you can head to the file menu and click on 'All Reservations'. This opens a new window in which the section for rendering the reservations is dynamically populated, which means it gets all the reservations from the database and displays it uniformly.



You might have noticed that each reservation has a mark as done button, when clicked, it marks the selected reservation as done, which removes it from our database and adds the table back to the list of available tables on the main screen.

The frameworks that were used in this application are :
Electron and Bootstrap

Electron is very lightweight and very powerful when it comes to cross platform development, it offers two-way data binding, which means data can be sent to/from the Frontend view to the back-end in one simple click without the need of complex architectures such as MVC.

In my application, electron is responsible for rendering HTML inside application, fetching and writing data to the database (in our case, local JSON files), sending data between different windows, making live changes to the UI as the user interacts with it.

Part of what makes Electron Apps a good alternative to a native desktop app is the fact that Electron apps behave like Web Apps. What sets them apart is that Web Apps can only download files to the computer's file system but Electron Apps can access the file system and can also read and write data, for example in our case, we are manipulating the JSON files that are stored in our file system.

But on the other hand, as electron is embedded with node and chromium, it is very resource intensive. It uses a lot memory for giving you a good fluid experience inside the app.

For the front-end, from the very beginning of production, I had finalized that I will be using bootstrap, because our application is cross-platform and it will be used on screens of different sizes, bootstrap makes it so much more efficient and responsive to window size changes as it keeps everything uniform with the help of media queries.

In our applicaiton, by default, the first window that opens up is set to be of exact height and width as the screen. The user can resize further if he wants.

Bootstrap offers premade javascript classes with very short names, these class names are easy to use but they can sometimes be very confusing.

The ability to arrange content on a page with bootstrap is very easy but it makes you forget other fundamentals such as flex-box, any avid bootstrap user might be scared to build something from scratch as everything is already half-cooked for him when using bootstrap.

In our case, the advantages outweigh the disadvantages so it was the first and last choice.

References:

[Food Delivery in Pakistan | Order Food Online on foodpanda](#)

[Hosting Platform of Choice](#)

[MockFlow - Online Wireframe Tools, Prototyping Tools, Online Whiteboard, Design tool, UI Mockups, UX Suite, Remote design collaboration, UX Planning](#)

[Bootstrap · The most popular HTML, CSS, and JS library in the world.](#)

www.chaicafe.ie

[Electron.js framework, its advantages and disadvantages - Framework](#)

[What is Bootstrap? Pros and Cons Of This Framework](#)