



YASAMAN FARSIAN SANI

ASSIGN SEATS TO PASSENGERS IN AN AIRCRAFT

SEKAL TECHNICAL INTERVIEW

Technologies

Front-End Part

- HTML
- SASS
- CSS
- Bootstrap5
- JavaScript



Back-End Part

- Node.js
- Express.js
- MYSQL

Front-End

HTML

For implementing the structure of the website

CSS, SASS

For Designing the HTML

BOOTSTRAP5

For having responsive design

JQUERY

for clicking function to get the which seat is selected
showing which seat is booked and adding disabled featured to input

Back-end

NODE.JS

For making a server , saving data in mysql database and and getting all the information from database

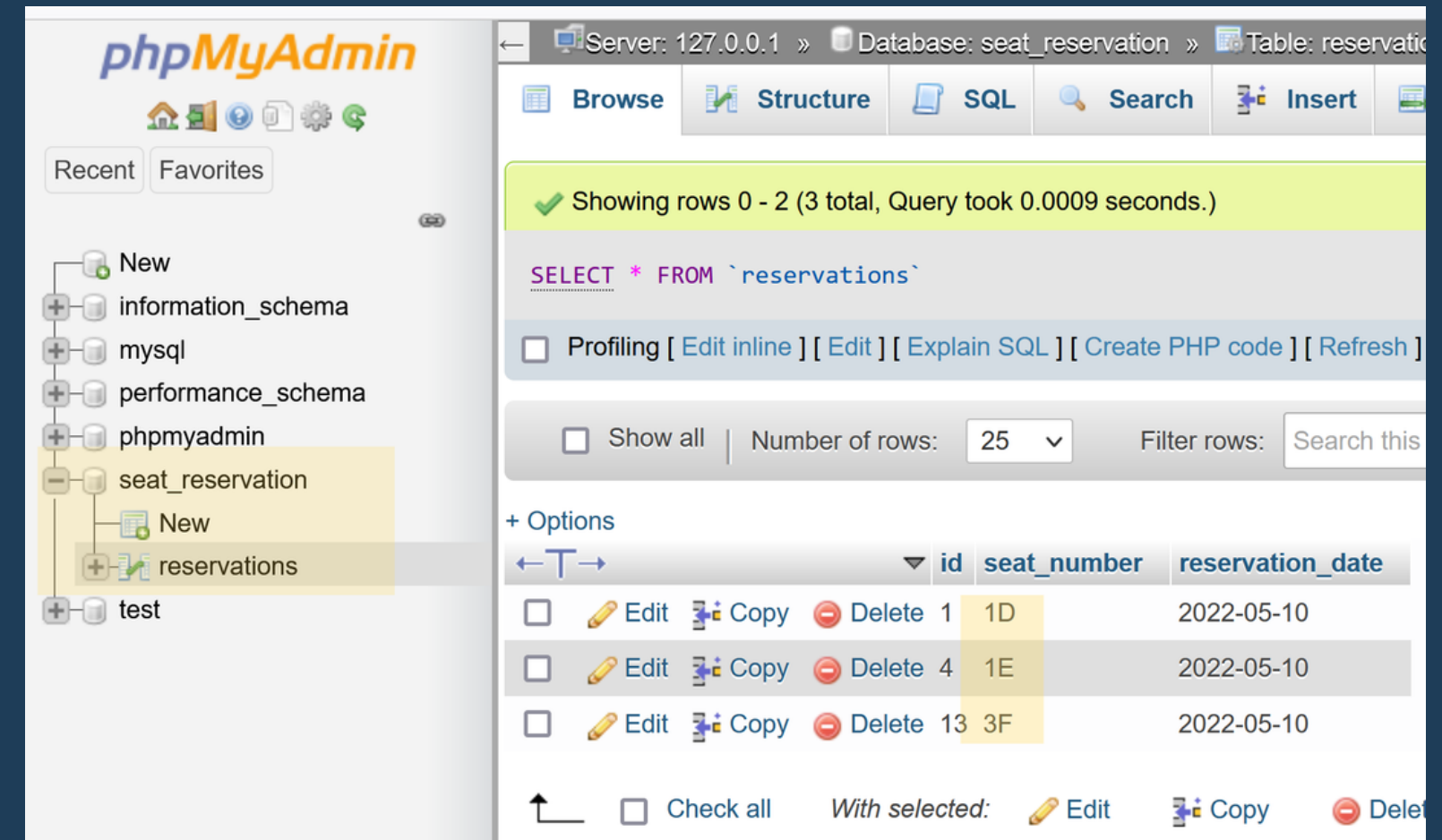
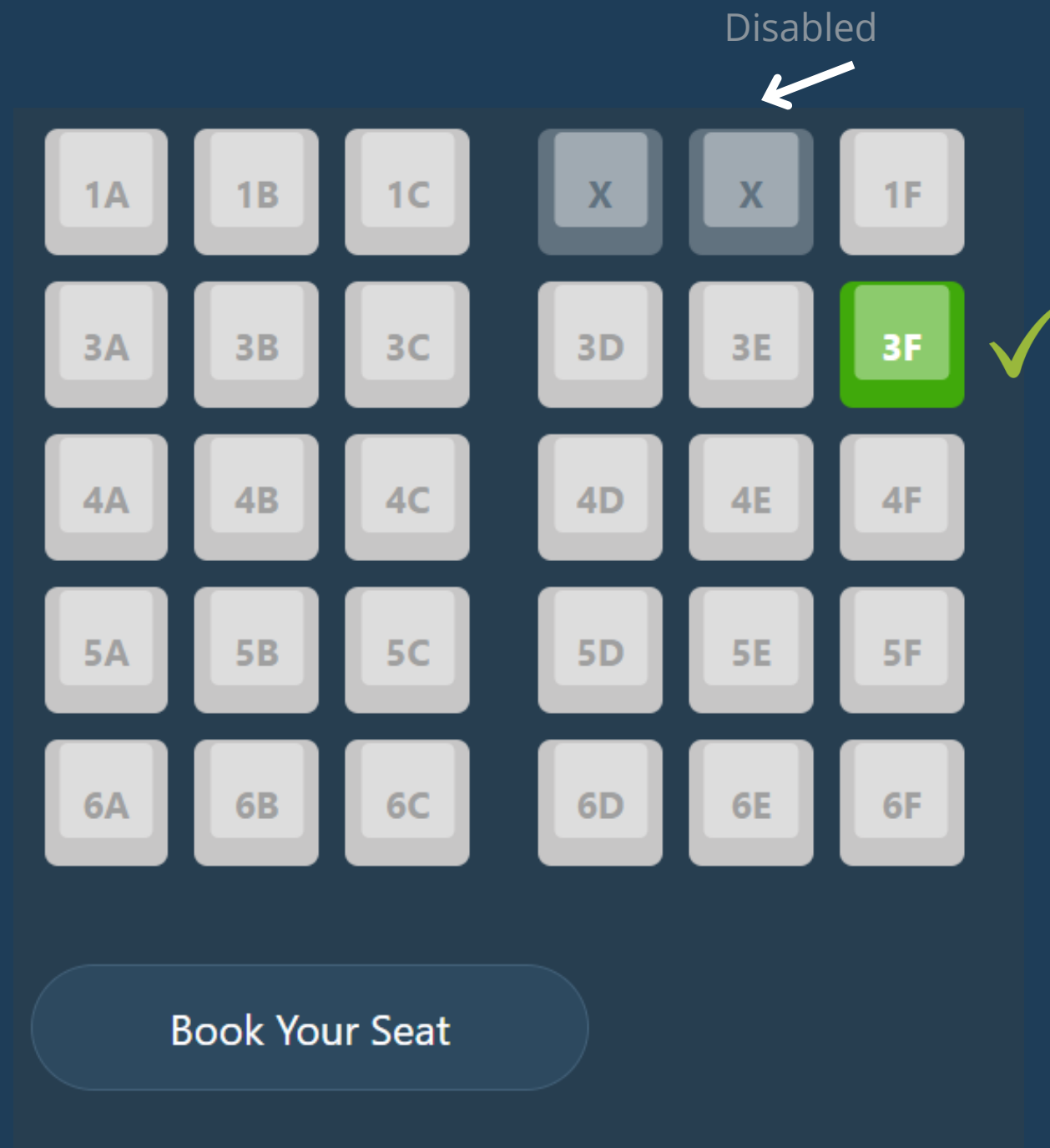
EXPRESS.JS

For making backend API

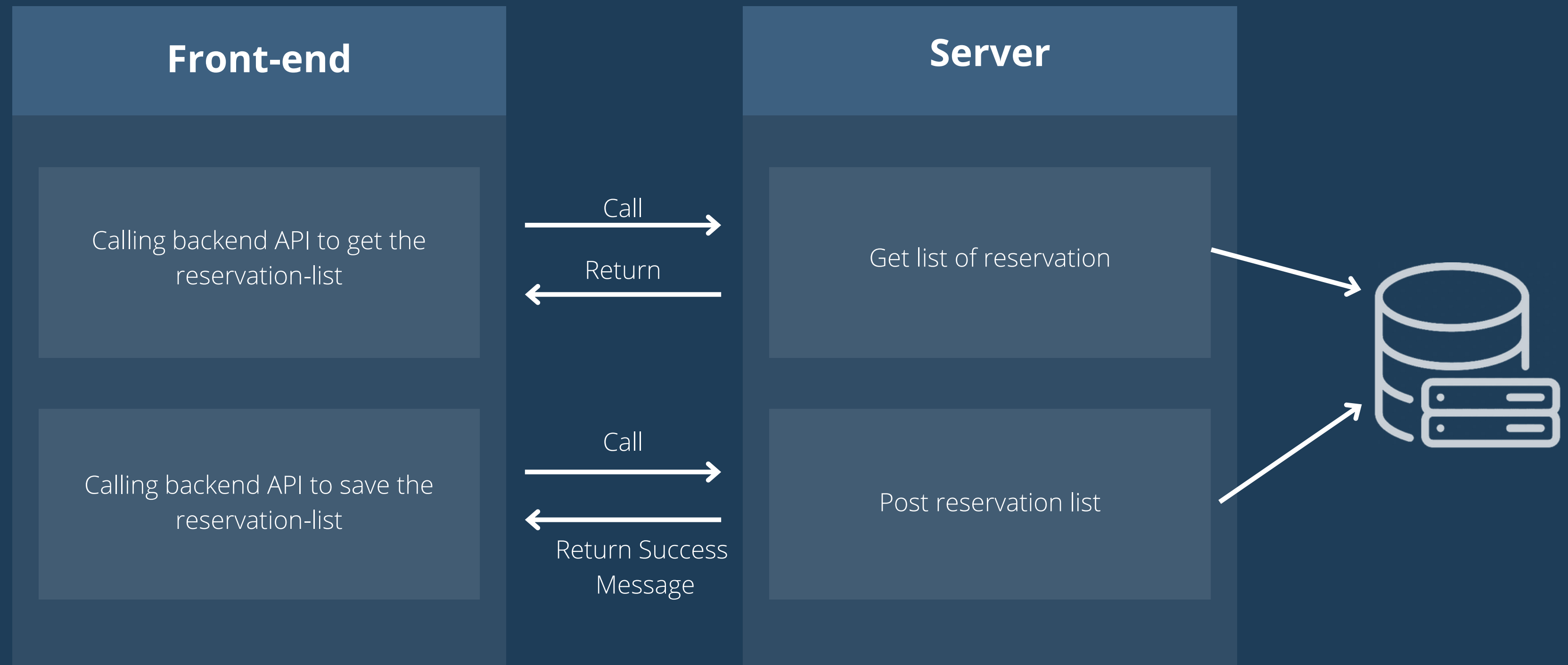
MYSQL

For creating Database

IDEA



DIAGRAM



CONFIG AND INSTALLATION

- creating a node server with express frame work:
- install npm
- install express --save
- install mysql to connect to the database
- install path npm package for reading from directory

```
const express = require('express');
const mysql = require('mysql');
const path = require('path')

const app = express();
const port = process.env.PORT || 8081;
app.use(express.json());

app.use(express.static("dist"));
```

IMPLEMENTING THE BACKEND

```
var myHeaders = new Headers();
myHeaders.append("Content-Type", "application/json");

fetch("http://localhost:8081/api/get-reserved-seats", {
  method: 'GET',
  headers: myHeaders
}).then(response => response.json())
.then(result => {
  $.each(result, function(index, value) {
    seat_number = value.seat_number;
    $(`input[name='seats[]'][value='${seat_number}']`).attr('disabled', 'disabled');
  });
}).catch(error => console.log('error', error));
```

Calling backend API to get the reservation-list

IMPLEMENTING THE BACKEND

```
$("#submit").click(function(e) {
  e.preventDefault();
  var bookingList = [];

  $('input:checked').each(function() {
    bookingList.push({
      'seat_number': $(this).val(),
      'reservation_date': "2022-05-10"
    });
  });

  console.log(bookingList)
  fetch("http://localhost:8081/api/book-reserved-seats", {
    method: 'POST',
    headers: myHeaders,
    body: JSON.stringify(bookingList),
    redirect: 'follow'
  }).then(response => response.text())
  .then(result => {
    console.log(result);
    location.reload();
  })
  .catch(error => console.log('error', error));
});
```

Calling backend API to save the reservation-list

IMPLEMENTING THE BACKEND

```
app.get('/api/get-reserved-seats', function(req, res) {  
  con.query("SELECT * FROM reservations", function(err, result, fields) {  
    if (err) throw err;  
    res.status(200).send(result);  
  });  
});
```

Get list of reservation

IMPLEMENTING THE BACKEND

```
app.post('/api/book-reserved-seats', function(req, res) {
  var sql = "INSERT INTO reservations (seat_number, reservation_date) VALUES ?";
  var value = [];

  for (var val of req.body) {
    list = [
      val.seat_number,
      val.reservation_date
    ]
    value.push(list);
  }

  con.query(sql, [value], function(err, result) {
    if (err) throw err;
    res.status(200).send({
      "message": "successfully inserted"
    });
  });
});
```

Post reservation list



FUTURE WORK

- Making better UI design
- Adding feature for cancelling the reservation of specific seats
- Showing successful / error message to user based on reservation process



RESOURCES

- <https://blog.logrocket.com/build-rest-api-node-express-mysql/>
- https://www.w3schools.com/nodejs/nodejs_mysql_insert.asp