**EASWARI ENGINEERING COLLEGE(AUTONOMOUS)**

**MINIPROJECTREPORT**

***Submittedby***

# NICHOLAS GAMLE J

***Inpartialfulfilmentfortheawardofthedegree***

***of***

# BACHELOROFTECHNOLOGY

***in***

## ARTIFICIALINTELLIGENCEANDDATASCIENCE

**EASWARIENGINEERINGCOLLEGE,CHENNAI600089**

# ANNAUNIVERSITY:CHENNAI600025

**ANNAUNIVERSITY:CHENNAI600025**

# BONAFIDECERTIFICATE

Certified that this project report“ **TO DO LIST ”**is the bonafide work of “**NICHOLAS GAMLE J**” who carried out the project work under my supervision.

SIGNATURE SIGNATURE

HEADOFTHEDEPARTMENTSUPERVISOR

**ARTIFICIAL INTELLIGENCE ANDDATASCIENCE**

**EASWARIENGINEERINGCOLLEGE, RAMAPURAM,CHENNAI**

**600 089**

**TO DO LIST**

**AIM:**

The aim of this project is to design and implement a to do list that has an user friendly interface and helps the user to prioritize their work and personal tasks .

## ALGORITHM:

1. Find the events which will be hosted in Izmir.
2. Find the event (event’s name, category name, organisers name) which was held with the highest attendance and the total obtained revenue.
3. Find the total number of events for each category in descending order.
4. Find the past events in which the members (whose age is between 18-25) participated.
5. Find the past events in which the total number of participating members is less than (not equal to) 3.
6. List the members who attended at least 3 events and update these members’ membership type to gold.
7. Delete the “academic” events which will be organised by Dokuz Eylül University on 19th May 2019.
8. Return the member’s email address who paid at most.
9. Update the discount rate of events which will be organised at cities whose name starts with "i" as 25.

10) Delete the members who have not participated in any events.

**IMPORTANT TERMS**

**RENDERING :**

Rendering is a process used in web development that turns website code into the interactive pages users see when they visit a website

**AXIOS:**

Axios is a popular JavaScript library for making HTTP requests from a web browser or a Node. Js server. It performs various operations, such as retrieving and sending data and handling responses.

**NEON POSTGRES :**

Neon is a fully managed serverless PostgreSQL with a generous free tier. Neon separates storage and compute, and offers modern developer features such as serverless, branching, bottomless storage, and more.

## PROGRAM:

## FRONT END :

## import { BrowserRouter, Routes, Route } from "react-router-dom";

## import './App.css'

## import 'bootstrap/dist/css/bootstrap.min.css';

## import axios from 'axios'

## axios.defaults.baseURL = "http://localhost:3000/";

## import Waves from "./assets/Waves/Waves"

## import Home from "./components/Home/Home.jsx"

## import Td from "./components/Td/Td.jsx";

## function App() {

## return (

## <>

## <BrowserRouter>

## <Routes>

## <Route path="/" element={<Home/>}/>

## <Route path="/td" element={<Td/>}/>

## </Routes>

## </BrowserRouter>

## <Waves/>

## </>

## )

## }

## export default App

## BACKEND:

## const express = require('express')

## const app = express()

## const cors = require('cors');

## const bodyParser = require('body-parser');

## const port = 3000

## const { Pool } = require('pg');

## require('dotenv').config();

## // -----------------Database Connection-----------------------

## const pool = new Pool({

## connectionString: process.env.DATABASE\_URL,

## ssl: {

## require: true,

## },

## });

## // ------------------------------------------------------------------

## app.use(bodyParser.json());

## app.use(cors());

## // ---------------------------------Routes-------------------------

## app.get('/', (req, res) => {

## res.send('home page')

## })

## //-------------------------------Todo Routes------------------------

## app.get('/td-:regid', async (req, res) => {

## const regid = req.params.regid;

## console.log("request recieved")

## try {

## const result = await pool.query(SELECT \* FROM tlist WHERE regid='${regid}');

## res.json(result.rows);

## } catch (error) {

## console.error('Error executing SQL:', error);

## res.status(500).send('Internal Server Error');

## }

## });

## app.post('/td-:regid', async (req, res) => {

## try {

## const regid = req.params.regid;

## const { title, content } = req.body;

## const query = 'INSERT INTO tlist (regid, title, content) VALUES ($1, $2, $3)';

## const values = [regid, title, content];

## const result = await pool.query(query, values);

## res.json(result.rows);

## } catch (error) {

## console.error('Error executing SQL:', error);

## res.status(500).send('Internal Server Error');

## }

## });

## 

## app.delete('/td-:regid', async (req, res) => {

## try {

## const regid = req.params.regid;

## const result = await pool.query('DELETE FROM tlist WHERE regid = $1', [regid]);

## // Check if rows were affected or not

## if (result.rowCount > 0) {

## res.status(200).json({ message: Deleted record with regid ${regid} });

## } else {

## res.status(404).json({ message: No record found with regid ${regid} });

## }

## } catch (error) {

## console.error('Error executing SQL:', error);

## res.status(500).send('Internal Server Error');

## }

## });

## // --------------------------------------------------------------------

## app.listen(port, () => {

## console.log(Example app listening on port ${port})

**DATABASE CONNECTION :**

const { Pool } = require('pg');

require('dotenv').config();

const pool = new Pool({

connectionString: process.env.DATABASE\_URL,

ssl: {

require: true,

},

});

module.export = pool;

**INPUT :**

**OUTPUT:**

**RESULT:**

In conclusion, the To-do list is executed successfully and helps the user to complete and organize the most crucial tasks and helps to improve time management