

## CODE

```
const express = require('express')
const app = express()
const port = 3000
const parcel = require('body-parser')
const mongo = require('mongoose')

//connecting mongodb
mongo.connect("mongodb+srv://Symbiote:cluster0@cluster0.b0xwi.mongodb.net/RIP?
retryWrites=true&w=majority")
  .then(() => console.log("DB Connected"))
  .catch((err) => {
    console.log(err)
    console.log("DB Connection Failed");
  });

//defining schema for question and answer key value pair
var listingSchema=mongo.Schema({
q:{type:String,required:true},
ans:{type:String}
});

//defining model name for questions answers insertion in database
var mod = mongo.model('cluster1',listingSchema);

//defining the authorization and features this app can use
app.use((req, res, next) => {
  res.setHeader("Access-Control-Allow-Headers", "Content-Type, Accept,
Authorization");
  res.setHeader("Access-Control-Allow-Origin", "*");
  res.setHeader("Access-Control-Allow-Methods", "GET, POST, OPTIONS, DELETE,
PUT");
  next();
});

app.use(parcel.urlencoded({ extended: false }));
app.use(parcel.json());

//mongo connection check
app.get('/listings',(req,res) => {
  mod.find().then((x) => {res.status(200).json(x) }).catch((err) => {
    console.log(err)
    console.log("DB Connection Failed");
  });
});

//trial
```

```

// practice API's
app.get('/list', (req, res) => {
  res.status(200).json({ message: "Hello world" });
})

app.get('/list1', (req, res) => {
  res.status(200).json({ name:"Rohan",address:"SP"});
})

app.post('/create', (req, res) => {
  res.status(200).json({ message: "Hello world" });
})

app.post('/sum', (req, res) => {
  console.log(req.body)
  const num= Number(req.body.num1) + Number(req.body.num2);
  res.status(200).json({ message: num });
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})

app.post('/data', (req, res) => {
  var sample= new mod({name:'abc',summary:'xyz'})
  var dummy = [{ name: 'Do you prefer cities or countryside?', summary: 'Countryside' }, { name: 'Do you prefer mountains or beach?', summary: 'Beach' }, { name: 'Do you prefer Hot or Cold weather?', summary: 'Cold' }]
  mod.create(dummy).then((x) => {res.status(200).json(x) }).catch((err) => {
    console.log(err)
    console.log("DB Connection Failed");
  });
})

//trial

//Code for selecting your destination
app.post('/locations', (req, res) => {
  var location="";
  console.log(req.body);
  mod.create(req.body).then((x) => {
    if(req.body[0].ans=="countryside"){
      if(req.body[1].ans=="mountains" && req.body[2].ans=="hot"){

```

```

        location="khuri";
    }
    if(req.body[1].ans=="mountains" && req.body[2].ans=="cold"){
        location="lehLadakh";
    }
    if(req.body[1].ans=="sea" && req.body[2].ans=="hot"){
        location="andamanAndNicobar";
    }
    if(req.body[1].ans=="sea" && req.body[2].ans=="cold"){
        location="lakshadweep";
    }
}
if(req.body[0].ans=="city"){
    if(req.body[1].ans=="mountains" && req.body[2].ans=="hot"){
        location="wayanad";
    }
    if(req.body[1].ans=="mountains" && req.body[2].ans=="cold"){
        location="sikkim";
    }
    if(req.body[1].ans=="sea" && req.body[2].ans=="hot"){
        location="goa";
    }
    if(req.body[1].ans=="sea" && req.body[2].ans=="cold"){
        location="pondicherry";
    }
}
res.status(200).json(location)
}).catch((err) => {
    console.log(err)
    console.log("DB Connection Failed");
});
})
//creating login schema with user and pass key value pair
var loginSchema=mongo.Schema({
    user:{type:String,required:true},
    pass:{type:String,required:true}
});

//defining model name for login data insertion in database
var mod2 = mongo.model('userlogin',loginSchema);

//login API
app.post('/login',(req,res) => {
    console.log(req.body)
    mod2.create(req.body).then((x) => {res.status(200).json({message:"User
Logged In Successfully!"}) }).catch((err) => {
        console.log(err)
    })
})

```

```
        console.log("DB Connection Failed");
    });
})
```

Data.services.ts (consists of all the endpoints)

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class DataService {
  random:boolean=false;

  constructor(private http: HttpClient) {

  }

  PostData(questions:any):Observable<any>{
    return this.http.post('http://localhost:3000/create',questions);
  }
  //practice
  SumData(numbers:any):Observable<any>{
    return this.http.post('http://localhost:3000/sum',numbers);
  }
  GetData():Observable<any>{
    return this.http.get('http://localhost:3000/listings');
  }

  TempData():Observable<any>{
    return this.http.post('http://localhost:3000/data',{});
  }
  GetTrips(data:any):Observable<any>{
    return this.http.post('http://localhost:3000/locations',data);
  }
  userLogin(data:any):Observable<any>{
    return this.http.post('http://localhost:3000/login',data);
  }
}
```

App.module.ts (consists of all the imports and paths and connections)

```

import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { AppRoutingModule } from './app-routing.module';
import { RouterModule } from '@angular/router';
import { AppComponent } from './app.component';
import { TrialComponent } from './trial/trial.component';
import { HttpClientModule } from '@angular/common/http';
import { BrowserAnimationsModule } from '@angular/platform-
browser/animations';
import { DreamComponent } from './dream/dream.component';
import { AboutUsComponent } from './about-us/about-us.component';
import { ContactUsComponent } from './contact-us/contact-us.component';
import { LoginComponent } from './login/login.component';
import { MainComponent } from './main/main.component';
import { FinalComponent } from './final/final.component';

@NgModule({
  declarations: [
    LoginComponent,
    AppComponent,
    TrialComponent,
    DreamComponent,
    MainComponent,
    FinalComponent
  ],
  imports: [
    FormsModule,ReactiveFormsModule,
    HttpClientModule,
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule,
    RouterModule.forRoot([
      {path: 'dream', component: DreamComponent},
      {path: 'about-us', component: AboutUsComponent},
      {path: 'contact-us', component: ContactUsComponent},
      {path: 'login', component: LoginComponent},
      {path:'main',component: MainComponent},
      {path:'final/:location',component:FinalComponent}
    ]),
  ],
  exports:[ RouterModule ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

## TOP MENU BAR(STATIC)

### CODE

Index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- Bootstrap CSS -->

  <title>Dream Destination Finder</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <link rel="preconnect" href="https://fonts.gstatic.com">
  <link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;500&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
  <script src="//code.jquery.com/jquery-1.11.3.min.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></sc
ript>

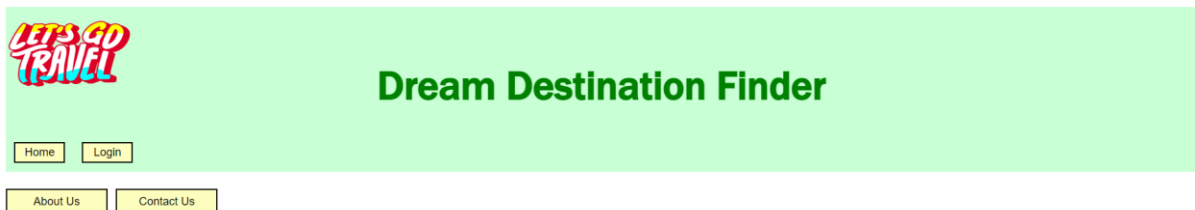
</head>
<body class="mat-typography">
  <app-root></app-root>

  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min
.js" integrity="sha384-
ka7Sk0GlN4gmtz2MlQnikT1wXgYs0g+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
</body>
</html>
```

## Styles.scss

```
.back
header{
width: 100%; height: 100vh;
background-color: white;
background-repeat: no-repeat;
background-size: cover;
}
nav{
width: 100%; height: 15vh;
display: flex; justify-content: space-between;
align-items: center;
}
nav .mainmenu{
gap: 20px;
display: flex; justify-content: space-around;
}
main{
width: 100%;
min-height: 85vh;
text-align: center;
}
section h3{
letter-spacing: 3px; font-weight: 200;
}
section h1{
text-transform: uppercase;
}
section div{
animation:changeborder 10s infinite linear;
border: 7px solid red;
}
```

## OUTPUT



## Homepage (DREAM)

## CODE

Dream.component.html

```

<div class="container1">
  <div class="background-wrap">
    <video autoplay [muted]="true" loop id="myVideo">
      <source src="/assets/video1.mp4" type="video/mp4">
      Your browser does not support HTML5 video.
    </video>
  </div>
  <section>
    <div class = "change_text" style="font-size:xx-large;"><b>Dream
Destination Finder</b></div><br>
    <p style="font-size: large;"><i>Your one stop site to find your next
Vacation </i></p><br>
    <input type = button class="btn btn-primary" value = "Find my perfect
Destination" routerLink="/main">
  </section>
</div>

```

Dream.component.scss

```

.back
header{
width: 100%; height: 100vh;
background-color:white;
background-repeat: no-repeat;
background-size: cover;
}

nav{
width: 100%; height: 15vh;
display: flex; justify-content: space-between;
align-items: center;
}
nav .mainmenu{
width: 40%;
display: flex; justify-content: space-around;
}
main{
width: 100%; height: 85vh;
text-align: center;
}
video{
width: 100%;
height: 100%;
object-fit: fill;
}
.background-wrap{
opacity: 0.5;
height: 100%;
}

```



```

}
.container1{
  position: relative;
  width: 100%;
  height: 100%;
}
section{
  position: absolute;
  top: 30vh;
  margin: auto;
  width: 100%;
  position: absolute;
}
section h3{
  letter-spacing: 3px; font-weight: 200;
}
section h1{
  text-transform: uppercase;
}
section div{
  animation:changeborder 10s infinite linear;
  border-style: groove;
  border: 7px lightgreen;
}

```

## Dream.component.ts

```

import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-dream',
  templateUrl: './dream.component.html',
  styleUrls: ['./dream.component.scss']
})
export class DreamComponent implements OnInit {

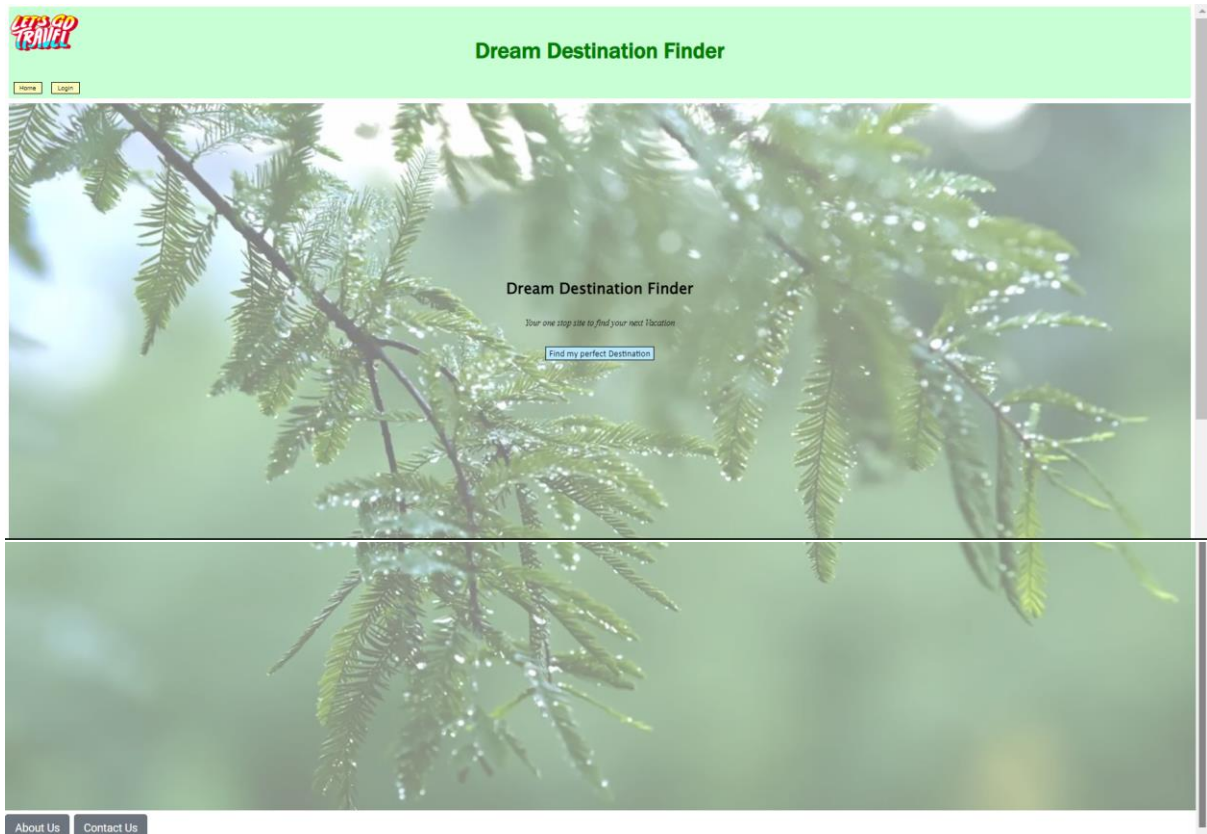
  constructor() { }

  ngOnInit(): void {
  }

}

```

## OUTPUT



## Login Page

### CODE

Login.component.html

```
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta charset="utf-8">
  <title></title>
  <script>
    function setFormMessage(formElement, type, message) {
      const messageElement = formElement.querySelector(".form__message");

      messageElement.textContent = message;
      messageElement.classList.remove("form__message--success", "form__message--error");
      messageElement.classList.add(`form__message--${type}`);
    }

    function setInputError(inputElement, message) {
      inputElement.classList.add("form__input--error");
      inputElement.parentElement.querySelector(".form__input-error-message").textContent = message;
    }
  </script>
</head>
```

```

function clearInputError(inputElement) {
    inputElement.classList.remove("form__input--error");
    inputElement.parentElement.querySelector(".form__input-error-
message").textContent = "";
}

document.addEventListener("DOMContentLoaded", () => {
    const loginForm = document.querySelector("#login");
    const createAccountForm = document.querySelector("#createAccount");

    document.querySelector("#linkCreateAccount").addEventListener("click", e
=> {
        e.preventDefault();
        loginForm.classList.add("form--hidden");
        createAccountForm.classList.remove("form--hidden");
    });

    document.querySelector("#linkLogin").addEventListener("click", e => {
        e.preventDefault();
        loginForm.classList.remove("form--hidden");
        createAccountForm.classList.add("form--hidden");
    });

    loginForm.addEventListener("submit", e => {
        e.preventDefault();

        // Perform your AJAX/Fetch login

        setFormMessage(loginForm, "error", "Invalid username/password
combination");
    });

    document.querySelectorAll(".form__input").forEach(inputElement => {
        inputElement.addEventListener("blur", e => {
            if (e.target.id === "signupUsername" && e.target.value.length > 0
&& e.target.value.length < 10) {
                setInputError(inputElement, "Username must be at least 10
characters in length");
            }
        });

        inputElement.addEventListener("input", e => {
            clearInputError(inputElement);
        });
    });
});
</script>
</head>

```

```

<div class="rapper">
<div class="container" *ngIf="showLogin;else successMessage">
    <h1 class="form__title">Login</h1>
    <div class="form__message form__message--error"></div>
    <div class="form__input-group">
        <input type="text" class="form__input" autofocus
placeholder="Username or email" [(ngModel)]="user">
        <div class="form__input-error-message"></div>
    </div>
    <div class="form__input-group">
        <input type="password" class="form__input" autofocus
placeholder="Password" [(ngModel)]="pass">
        <div class="form__input-error-message"></div>
    </div>
    <button class="btn btn-primary" type="submit"
(click)="userLogin()">Continue</button><br>
    </div>
</div>
<ng-template #successMessage>
    <h1 class="form__title">User Logged In Successfully!</h1>
</ng-template>
<script src="C://1-CODINGS//Jcomp//main.js"></script>

```

## Login.component.scss

```

body {
    --color-primary: #009579;
    --color-primary-dark: #007f67;
    --color-secondary: #252c6a;
    --color-error: #cc3333;
    --color-success: #4bb544;
    --border-radius: 4px;

    margin: 0;
    height: 100vh;
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 18px;
    background-size: cover;
}

.container {
    width: 400px;
    max-width: 400px;
}

```

```
margin: 1rem;
padding: 2rem;
box-shadow: 0 0 40px rgba(0, 0, 0, 0.2);
border-radius: var(--border-radius);
background: #ffffff;
align-items: center;
align-content: center;
}

.container,
.form__input,
.form__button {
  font: 500 1rem 'Quicksand', sans-serif;
}

.form--hidden {
  display: none;
}

.form > *:first-child {
  margin-top: 0;
}

.form > *:last-child {
  margin-bottom: 0;
}

.form__title {
  margin-bottom: 2rem;
  text-align: center;
}

.form__message {
  text-align: center;
  margin-bottom: 1rem;
}

.form__message--success {
  color: var(--color-success);
}

.form__message--error {
  color: var(--color-error);
}

.form__input-group {
  margin-bottom: 1rem;
}
```

```
.form__input {
  display: block;
  width: 100%;
  padding: 0.75rem;
  box-sizing: border-box;
  border-radius: var(--border-radius);
  border: 1px solid #dddddd;
  outline: none;
  background: #eeeeee;
  transition: background 0.2s, border-color 0.2s;
}

.form__input:focus {
  border-color: var(--color-primary);
  background: #ffffff;
}

.form__input--error {
  color: var(--color-error);
  border-color: var(--color-error);
}

.form__input-error-message {
  margin-top: 0.5rem;
  font-size: 0.85rem;
  color: var(--color-error);
}

.form__button {
  width: 100%;
  padding: 1rem 2rem;
  font-weight: bold;
  font-size: 1.1rem;
  color: #ffffff;
  border: none;
  border-radius: var(--border-radius);
  outline: none;
  cursor: pointer;
  background: var(--color-primary);
}

.form__button:hover {
  background: var(--color-primary-dark);
}

.form__button:active {
  transform: scale(0.98);
}
```

```

}

.form__text {
  text-align: center;
}

.form__link {
  color: var(--color-secondary);
  text-decoration: none;
  cursor: pointer;
}

.form__link:hover {
  text-decoration: underline;
}

.rapper{
  width: 100vw;
  height: 80vh;
  display: flex;
  justify-content: center;
  align-items: center;
}

```

## Login.component.ts

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { DataService } from '../data.service';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.scss']
})
export class LoginComponent implements OnInit {
  user:any;
  pass:any;
  showLogin=true;
  constructor(private data:DataService,private router:Router) {

  }

  ngOnInit(): void {
  }

  userLogin(){
    this.data.userLogin({user:this.user,pass:this.pass}).subscribe(res => {
      if(res.message.includes("Success"))

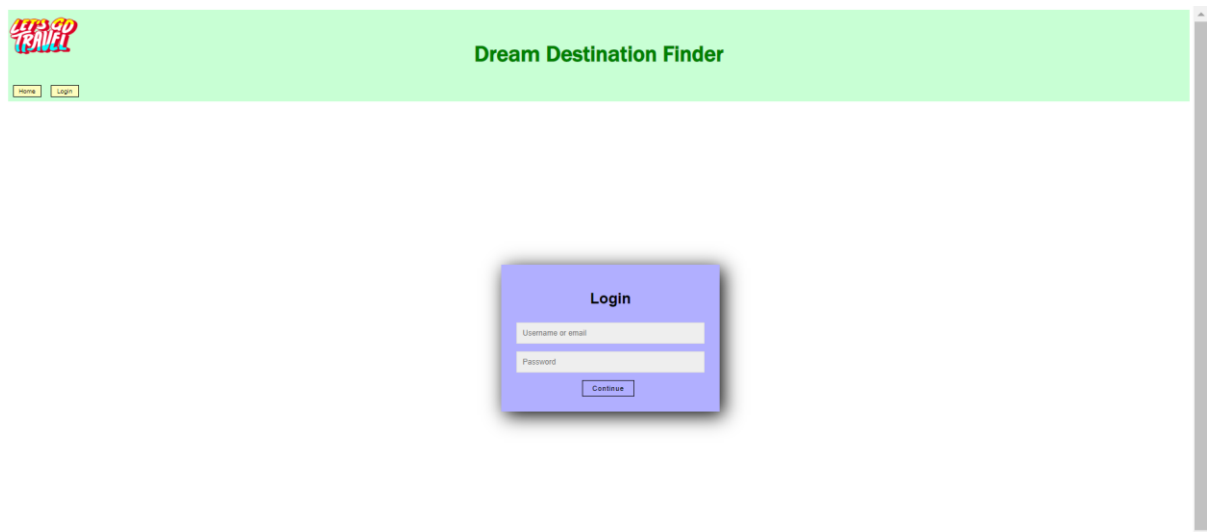
```

```

    this.showLogin=false;
    setTimeout(()=>{
      this.router.navigate(["/dream"])
    },3000)
  })
}
}

```

## OUTPUT



## Main Page

## CODE

Main.component.html

```

<div class="quiz-container" id="quiz">
  <div class="quiz-header">
    <h2 id="question">{{question.question1}}</h2>
    <ul>
      <li>
        <input [(ngModel)]="answer.answer1" type="radio"
class="answer" name="q1" value="city">
        <label id="a_text_q1">City</label>
      </li>
      <li>
        <input [(ngModel)]="answer.answer1" type="radio"
class="answer" name="q1" value="countryside">
        <label id="b_text_q1">Countryside</label>
      </li>
    </ul>
  </div>

```



```

        <div class="quiz-header">
            <h2 id="question">{{question.question2}}</h2>
            <ul>
                <li>
                    <input [(ngModel)]="answer.answer2" type="radio"
class="answer" name="q2" value="mountains">
                    <label id="a_text_q1">Mountains</label>
                </li>
                <li>
                    <input [(ngModel)]="answer.answer2" type="radio"
class="answer" name="q2" value="sea">
                    <label id="b_text_q1">Sea</label>
                </li>
            </ul>
        </div>
        <div class="quiz-header">
            <h2 id="question">{{question.question3}}</h2>
            <ul>
                <li>
                    <input [(ngModel)]="answer.answer3" type="radio"
class="answer" name="q3" value="hot">
                    <label id="a_text_q1">Hot</label>
                </li>
                <li>
                    <input [(ngModel)]="answer.answer3" type="radio"
class="answer" name="q3" value="cold">
                    <label id="b_text_q1">Cold</label>
                </li>
            </ul>
        </div>
    </div>
    <button type="submit" class="btn btn-primary" id="submit"
(click)="onSubmit()">Submit</button>

```

## Main.component.scss

```

@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500&dis
play=swap');

*{
    box-sizing: border-box;
}
body{
    background-color: #b8c6db;
    background-image: linear-gradient(315deg, #b8c6db 0%, #f5f7f7 100%);
    font-family: 'Poppins', sans-serif;
    display: flex;

```

```
    align-items: center;
    justify-content: center;
    height: 100vh;
    overflow: hidden;
    margin: 0;
}
.quiz-container{
    overflow: hidden;
    display: grid;
    justify-content: center;
}
.quiz-header{
    width: 50vw;
    background-color: #fff;
    border-radius: 10px;
    box-shadow: 0 0 10px 2px rgba(100, 100, 100, 0.1);
    padding: 4rem;
}
h2{
    padding: 1rem;
    text-align: center;
    margin: 0;
}

ul{
    list-style-type: none;
    padding: 0;
}
ul li{
    font-size: 1.2rem;
    margin: 1rem 0;
}
ul li label{
    cursor: pointer;
}
button{
    background-color: #03cae4;
    color: #fff;
    border: none;
    display: block;
    width: 100%;
    cursor: pointer;
    font-size: 1.1rem;
    font-family: inherit;
    padding: 1.3rem;
}
button:hover{
    background-color: #04adc4;
```

```

}
button:focus{
  outline: none;
  background-color: #44b927;
}

```

## Main.component.ts

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { DataService } from '../data.service';

@Component({
  selector: 'app-main',
  templateUrl: './main.component.html',
  styleUrls: ['./main.component.scss']
})
export class MainComponent implements OnInit {
  answer={
    answer1:null,
    answer2:null,
    answer3:null,
  }
  question={
    question1:"Do you prefer City or Countryside?",
    question2:"Do you prefer Mountains or Sea?",
    question3:"Do you prefer Hot or Cold weather?",
  }

  constructor(private dataService:DataService, private router:Router) {
  }

  ngOnInit(): void {
  }

  onSubmit(){
    const request=[{
      q:this.question.question1, ans:this.answer.answer1,
    },
    {
      q:this.question.question2, ans:this.answer.answer2,
    },
    {
      q:this.question.question3, ans:this.answer.answer3,
    }
  ]
    this.dataService.GetTrips(request).subscribe(response => {
      if(response)
        this.router.navigate(["/final/"+response])
    })
  }
}

```

```
});

}

}
```

## OUTPUT

## Final Page

## CODE

Final.component.html

```
<div *ngIf="currentLocation; else noLocation">
<div class="row">
  <div class="col-12">
    <div class="header">
      <img src='{{"/assets/"+currentLocation.name+".jpg"}}'
class="image" />
      <div class="title">
        {{currentLocation.label}}
      </div>
    </div>
  </div>
</div>

<br><br>
<div class="row">
  <div class="col-12">
    <div class="desc">
      <div class="name">
```

```

        {{currentLocation.label}}
    </div>

    <div class="para">
        {{currentLocation.desc}}
    </div>
</div>

<div class="name">
    Nearby Attractions
</div>

<br><br>
<div class="row">
    <div class="col-4" *ngFor="item of [1, 2, 3]; let i=index">
        <img
src='{{"/assets/"+currentLocation.name+"_"+(i+1)+".jpg"}}' style="width:
400px;height: 400px;" />
    </div>

</div>

</div>
</div>
</div>
<ng-template #noLocation>
<div class="name">
    No Location Found
</div>
</ng-template>

```

## Final.component.scss

```

.image{
    height: 70vh;
    width: 100vw;
}
.header{
    position: relative;
}
.title{
    position: absolute;
    bottom: 30px;
    font-size: 36px;
}

```

```

    font-weight: bold;
    color: white;
    padding: 0 10vw;
  }
  .desc{
    padding: 0 10vw;
    margin-top: 64px;
  }
  .name{
    font-size: 28px;
    font-weight: bold;
  }
  .para{
    margin-top: 24px;
    font-size: 20px;
    margin-bottom: 64px;
  }
}

```

## Final.component.ts

```

import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';

@Component({
  selector: 'app-final',
  templateUrl: './final.component.html',
  styleUrls: ['./final.component.scss']
})
export class FinalComponent implements OnInit {
  locations=[
    {name:"khuri",label:"Khuri Hills, Jaisalmer",desc:"Famous for its sand dunes, Khuri is a small village about 50 km south-west of Jaisalmer in Rajasthan. It is the perfect place to enjoy the sunrise, sunset as well as the delicious Rajasthani Cuisine! The camel rides in Kuri are a great attraction as well."},
    {name:"lehLadakh",label:"Leh-Ladakh",desc:"A surreal landscape surrounded by the sky-piercing peaks of the snow-capped Himalayan, Zaskar and Karakoram ranges, and adorned with crystal waters of the Shyok, Zaskar and Indus rivers, Leh, in Jammu and Kashmir, is the main city in the Ladakh region, one of the coldest deserts in the world. Perched at a height of over 11,000 ft above sea level, the city of Leh is a haven of adventure sports. Its rugged terrain and gushing rivers provide ripe opportunities for activities like trekking, river rafting, camping, mountain climbing and biking. As the winter months approach, the city is enveloped in a blanket of snow, making it a paradisiacal setting for winter sports. The renowned Chadar Trek starts from the village of Chilling, 66 km away, and mostly covers the frozen area of

```

River Zaskar. Throughout the adventure, trekkers are walking on precarious icy formations, staying in caves filled with stalactites and stalagmites and encountering other snowy landscape."},

{name:"andamanAndNicobar",label:"Andaman and Nicobar Islands",desc:"With turquoise waters and shimmering white beaches surrounded by a mangrove forest and primitive jungles, the Andaman and Nicobar Islands are a tropical paradise that draw visitors from far-flung places."},

{name:"lakshadweep",label:"Lakshadweep Islands",desc:"Lakshadweep is one of the worlds most spectacular tropical island systems. Thirty- two sq. km of land spread over 36 islands surrounded by 4200 sq. km of lagoon rich in marine wealth. The precious heritage of ecology and culture is supported by an extremely fragile ecosystem. Committed to the cause of Eco tourism Union Territory of Lakshadweep has consciously followed a middle path between tourism promotion and environmental conservation."},

{name:"wayanad",label:"Wayanad, Kerala",desc:"This popular town is famous for wildlife and its spice plantations. Wayanad is part of a forest reserve, located on the border of Tamil Nadu and Kerala. Because of it wild green coverage it offers great scenic beauty along with the experience of seeing wildlife. The weather here is mostly constant but October till May are the best months to visit. It is located at a distance of 76 km from the sea shores of Kozhikode. Wayanad hills are contiguous to Mudumalai in Tamil Nadu and Bandipur in Karnataka, thus forming a vast land mass for the wild life to move about in their most natural abode."},

{name:"sikkim",label:"Sikkim",desc:"Sikkim situated in the Eastern Himalayan region with an area of 7,096 sq km and altitude ranging from 300m to 8,586m above sea level is a wonderland blessed with natural abundance from icy cold deserts, flowering alpine meadows, to lush green forests and emerald mountain lakes. The crowning glory of this wonderland is undoubtedly the magnificent Mt. Khangchendzonga (8,586m), the third highest mountain of the world. The mountain is not merely a physical entity but the abode of guardian deity whose benign watchfulness ensures peace and prosperity of the land."},

{name:"goa",label:"Goa",desc:"Goa, situated on the west coast of India, is one of the most delightful states of India. Formerly a Portuguese colony, it is endowed with a variety of attractions including palm fringed beaches, miles of golden sands, lush green country-side, an incredible mosaic of cultural heritage, magnificent churches, temples, forts and monuments and a unique cultural synthesis of the east and west. With its tropical climate, Goa is a tourist destination for all seasons. Goa is a treasure trove of culture, music, dances and art forms that can be enjoyed during local festivals and celebrations. Animal lovers can visit the State's wildlife sanctuaries It is often said that 'Goa never stops partying'. The city is known for its vibrancy that attains its pinnacle in the annual Goa Carnival festival."},

{name:"pondicherry",label:"Pondicherry",desc:"A French colony till 1954, Puducherry (earlier Pondicherry) is a small Union territory in India that retains a distinctly French ambience and culture. It has a spiritual ambience, unspoilt stretch of beaches, and backwaters along with a surprising mix of cuisines. Every year, it attracts thousands of tourists who come here to visit the Aurobindo Ashram, meditate and learn yoga. Food, ocean view, coconut

trees, clean air, and peaceful shores, this place has a color for everyone and is definitely a great weekend getaway from Chennai or Bangalore."}

```
];  
currentLocation:any;  
  
constructor(private route:ActivatedRoute) { }  
  
ngOnInit(): void {  
  this.route.params.subscribe(param => {  
    if(param['location'])  
      this.getCurrentLocation(param['location']);  
  })  
}  
getCurrentLocation(param:string){  
  const loc= this.locations.filter(loc => loc.name==param)  
  if(loc.length>0)  
    this.currentLocation=loc[0];  
}  
}
```



## OUTPUT

(Showing 1 output out of possible 8)

Options chosen- CITY, MOUNTAINS, COLD

Suggested Place- Sikkim



### Sikkim

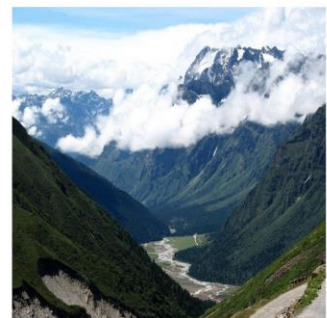
Sikkim situated in the Eastern Himalayan region with an area of 7,096 sq km and altitude ranging from 300m to 8,586m above sea level is a wonderland blessed with natural abundance from icy cold deserts, flowering alpine meadows, to lush green forests and emerald mountain lakes. The crowning glory of this wonderland is undoubtedly the magnificent Mt. Khangchendzonga (8,586m), the third highest mountain of the world. The mountain is not merely a physical entity but the abode of guardian deity whose benign watchfulness ensures peace and prosperity of the land.

### Nearby Attractions



About Us

Contact Us



## About US page

### CODE

About-us.component.html

```
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<div class="about-section">
  <h1>About This Project</h1>
  <p>I have written around 2000 lines of code by hand.</p>
  <p>I studied everything from scratch and spent 30+ hours just studying and learning how to use the tools for this project.</p>
  <p>I have used Angular to build the web application</p>
  <p>HTML + CSS + JavaScript + Bootstrap for web page designing</p>
</div>
```

```

    <p>NodeJs as Local API Host, Json for parcing files from FrontEnd to BackEnd
and </p>
    <p>MongoDB (NoSQL) as our Database. </p>
</div>

<h2 style="text-align: center;">Our Team</h2>
<div class="row">
    <div class="col-6">
        <div >
            
            <div >
                <h2>Rohan Pawar</h2>
                <p>RegNo.-20BAI1201</p>
                <p>Computer Science Engineerng with specialization in Artificial
Intelligence and Machine Learning</p>
                <p>22rohan.pawar@gmail.com</p>
            </div>
        </div>
    </div>

    <div class="col-6">
        <div>
            
            <div>
                <h2>Sri Ganesh Raj</h2>
                <p>Reg.No-20BAI1181</p>
                <p>Computer Science Engineerng with specialization in Artificial
Intelligence and Machine Learning</p>
                <p>sgr4051@gmail.com</p>
            </div>
        </div>
    </div>
</div>

```

### About-us.component.scss

```

body {
    font-family: Arial, Helvetica, sans-serif;
    margin: 0;
}

html {
    box-sizing: border-box;
}

```

```
*, *:before, *:after {
  box-sizing: inherit;
}

.column1 {
  float: left;
  width: 33.3%;
  margin-bottom: 16px;
  padding: 0 8px;
}

.column2 {
  float: right;
  width: 33.3%;
  margin-left: 66.6%;
  padding: 0 8px;
}

.about-section {
  padding: 50px;
  text-align: center;
  background-color: #474e5d;
  color: white;
}

.container {
  padding: 0 16px;
}

.container::after, .row::after {
  content: "";
  clear: both;
  display: table;
}

.title {
  color: grey;
}

.button {
  border: none;
  outline: 0;
  display: inline-block;
  padding: 8px;
  color: white;
  background-color: #000;
  text-align: center;
  cursor: pointer;
}
```

```
width: 100%;
}

.button:hover {
  background-color: #555;
}

@media screen and (max-width: 650px) {
  .column {
    width: 100%;
    display: block;
  }
}
```

## About-us.component.ts

```
import { Component, OnInit } from '@angular/core';

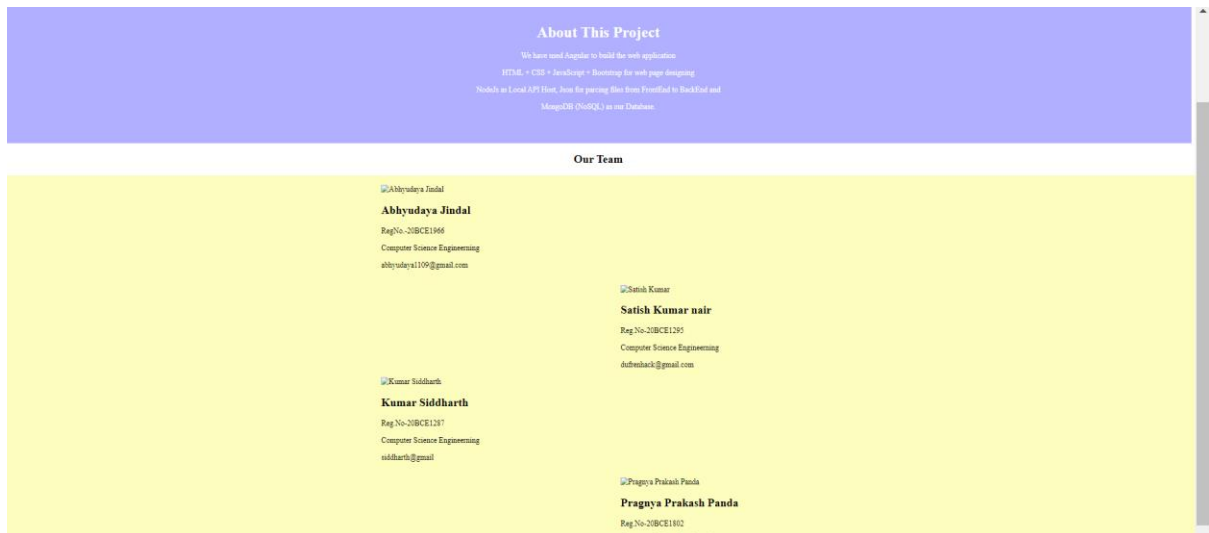
@Component({
  selector: 'app-about-us',
  templateUrl: './about-us.component.html',
  styleUrls: ['./about-us.component.scss']
})
export class AboutUsComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

}
```

## OUTPUT



## Contact US Page

### CODE

Contact-us.component.html

```
<head>
  <title></title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Lato">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
</head>

<div class="w3-content w3-container w3-padding-64" id="contact">
  <p class="w3-center"><em>We'd love your feedback!</em></p>
  <div class="w3-row w3-padding-32 w3-section">
    <div class="w3-col m4 w3-container">
      
    </div>
    <div class="w3-col m8 w3-panel">
      <div class="w3-large w3-margin-bottom">
        <i class="fa fa-map-marker fa-fw w3-hover-text-black w3-xlarge w3-
margin-right"></i> VIT, Chennai<br><br>
        <i class="fa fa-phone fa-fw w3-hover-text-black w3-xlarge w3-margin-
right"></i> Phone: 9993116448 / 8939108313<br><br>
        <i class="fa fa-envelope fa-fw w3-hover-text-black w3-xlarge w3-
margin-right"></i> Email: 22rohan.pawar@gmail.com<br><br>
      </div>
      <form action="/action_page.php" target="_blank">
        <div class="w3-row-padding" style="margin:0 -16px 8px -16px">
```

```

        <div class="w3-half">
            <input class="w3-input w3-border" type="text" placeholder="Name"
required name="Name">
        </div>
        <div class="w3-half">
            <br>
            <input class="w3-input w3-border" type="text"
placeholder="Email" required name="Email">
        </div>
    </div>
    <br>
    <input class="w3-input w3-border" type="text" placeholder="Message"
required name="Message">
    <button class="w3-button w3-black w3-right w3-section"
type="submit">
        <i class="fa fa-paper-plane"></i> SEND MESSAGE
    </button>
</form>
</div>
</div>
</div>

```

## Contact-us.component.scss

```

body,h1,h2,h3,h4,h5,h6 {font-family: "Lato", sans-serif;}
body, html {
    height: 100%;
    color: #777;
    line-height: 1.8;
}

/* Create a Parallax Effect */
.bgimg-1, .bgimg-2, .bgimg-3 {
    background-attachment: fixed;
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;
}

/* First image (Logo. Full height) */
.bgimg-1 {
    background-image: url('/w3images/parallax1.jpg');
    min-height: 100%;
}

/* Second image (Portfolio) */

```

```

.bgimg-2 {
  background-image: url("/w3images/parallax2.jpg");
  min-height: 400px;
}

/* Third image (Contact) */
.bgimg-3 {
  background-image: url("/w3images/parallax3.jpg");
  min-height: 400px;
}

.w3-wide {letter-spacing: 10px;}
.w3-hover-opacity {cursor: pointer;}

/* Turn off parallax scrolling for tablets and phones */
@media only screen and (max-device-width: 1600px) {
  .bgimg-1, .bgimg-2, .bgimg-3 {
    background-attachment: scroll;
    min-height: 400px;
  }
}

```

## Contact-us.component.ts

```

import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-contact-us',
  templateUrl: './contact-us.component.html',
  styleUrls: ['./contact-us.component.scss']
})
export class ContactUsComponent implements OnInit {

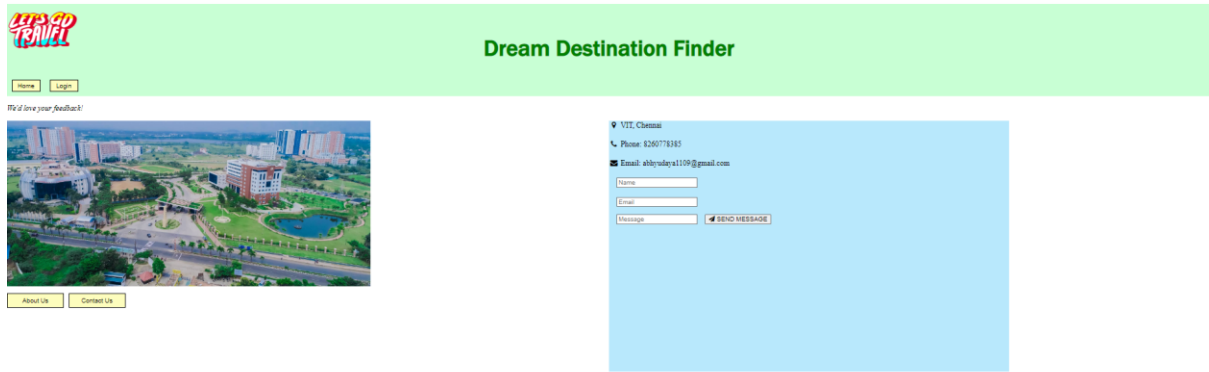
  constructor() { }

  ngOnInit(): void {
  }

}

```

## OUTPUT

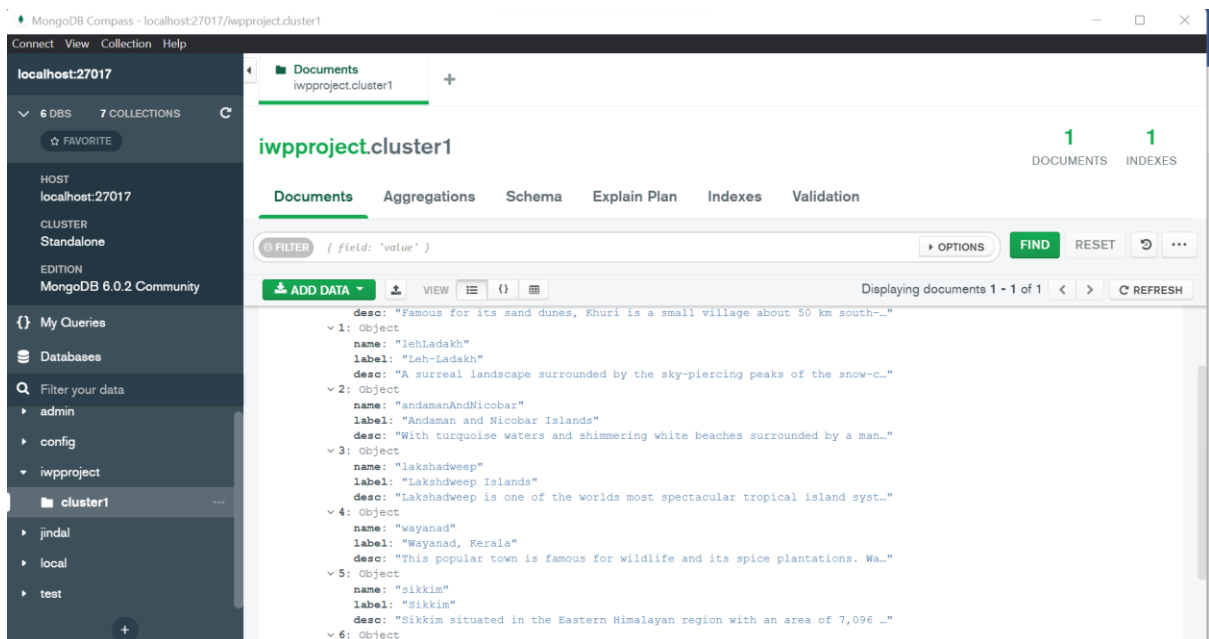


## DATABASE

Database name- RIP

Table Name- cluster1

Consists of all the answers to the questionnaire that you have selected and gets updated in real-time synchronously.





Example app listening on port 3000

DB Connected

```
[
  { q: 'Do you prefer City or Countryside?', ans: 'city' },
  { q: 'Do you prefer Mountains or Sea?', ans: 'mountains' },
  { q: 'Do you prefer Hot or Cold weather?', ans: 'cold' }
]
{ user: 'abhyudaya', pass: 'abhyudaya' }
[
  { q: 'Do you prefer City or Countryside?', ans: 'city' },
  { q: 'Do you prefer Mountains or Sea?', ans: 'mountains' },
  { q: 'Do you prefer Hot or Cold weather?', ans: 'cold' }
]
```