CAMPUS NAVIGATION SYSTEM

Problem Statement:~

Navigating college campuses, especially for newcomers and during events, poses significant challenges due to their intricate layouts and reliance on outdated static maps. Current solutions lack real-time updates, resulting in frustrations and inefficiencies for users. There is a pressing need for a new campus navigation system that offers dynamic, customized routes tailored to address issues like construction, accessibility concerns, and individual preferences. Such a system would significantly enhance the overall campus navigation experience, ensuring smoother and more efficient journeys for all users.

Modularization Of Problem Statement:~

Modularization for campus navigation involves breaking down the navigation system into distinct, manageable modules. These modules could cover mapping, route planning, user interface, localization, and data management. Each module focuses on a specific aspect, allowing for easier development, maintenance, and integration. This approach promotes scalability and flexibility, enabling seamless updates or additions to the navigation system. Ultimately, modularization enhances efficiency and user experience in navigating complex campus environments.

1. Map Display Module:~

- **Description:** Displays a map of the campus or area using a mapping service like Google Maps or Leaflet.js. It shows key landmarks, buildings, and pathways within the campus.
- **Components:**

  Integration with mapping service API
  Rendering of campus landmarks and pathways
  User interface for map interaction

2. User Location Detection Module:~

- **Description:** Utilizes the Geolocation API to detect the user's current location automatically when they access the application.
- **Components:**

  Geolocation API integration
  User permission handling
  Real-time location updating

CAMPUS NAVIGATION SYSTEM

3. Destination Selection Module:~

- **Description:** Allows users to select their destination within the campus by clicking on the map or selecting from a list of predefined destinations.
- **Components:**

  Destination input interface
  Map interaction for destination selection
  Destination list management

4. Step-by-Step Directions Module:~

- **Description:** Provides users with step-by-step directions on how to reach their destination, including landmarks or key points to look out for along the way.
- **Components:**

  Route calculation algorithm
  Turn-by-turn direction rendering
  Landmark and point-of-interest identification

5. Search Functionality Module:~

- **Description:** Implements a search feature that allows users to search for specific buildings, facilities, or landmarks within the campus and get directions to them.
- **Components:**

  Search bar interface
  Backend search algorithm
  Integration with direction module for route calculation

6. Save Favorite Locations Module:~

- **Description:** Allows users to save their favorite locations or frequently visited spots within the campus for quick access in the future.
- **Components:**

  Favorite locations management interface
  Backend storage for saved locations
  Quick access functionality

7. Active Users Location Module:~

- **Description:** Shows the active users' location to the administration.
- **Components:**

  Real-time tracking functionality
  Privacy controls for user consent
  Admin interface for location visualization

8. Location Information Module:~

- **Description:** Displays the information of the selected location along with images and coordinates.
- **Components:**

  Location details retrieval from database or API
  Image gallery integration
  Coordinate display

9. Feedback Module:~

- **Description:** Updates the application based on the given feedback.
- **Components:**

  Feedback submission interface
  Backend processing for feedback
  Implementation of feedback-driven updates

Design Methodology:~

To create an effective campus navigation methodology, start with a thorough survey of the layout, including buildings, pathways, and landmarks. Digitize this data for integration into a mapping platform, ensuring easy visualization. Refine the platform based on user feedback and testing, ensuring accuracy and usability. Enhance navigation with GPS and augmented reality technologies. Lastly, prioritize ongoing maintenance and updates to adapt to campus changes and improve user experience. This approach ensures intuitive, efficient, and accessible navigation for all campus stakeholders.

1. Requirement Analysis:

- Gather requirements from stakeholders including students, faculty, administration, and visitors.
- Identify key features and functionalities based on user needs and pin points.

2. User Experience (UX) Design:

- Develop user personas representing different user types and their navigation preferences.
- Design intuitive user interfaces for map display, destination selection, and search functionality.
- Create wireframes and prototypes to visualize the user journey and interaction flow.

CAMPUS NAVIGATION SYSTEM

3. Map Integration:

- Select a mapping service such as Google Maps or Leaflet.js for displaying campus maps.
- Customize map displays to include key landmarks, buildings, pathways, and points of interest (POIs).

4. Geolocation Integration:

- Implement Geolocation API to automatically detect the user's current location upon accessing the application.
- Ensure privacy and security measures are in place for handling user location data.

5. Destination Selection:

- Enable users to select their destination within the campus through map interaction or a list of predefined locations.
- Implement intuitive controls for users to easily choose their desired destination.

6. Routing Algorithm:

- Develop a routing algorithm to generate step-by-step directions from the user's current location to their selected destination.
- Incorporate real-time data on construction, closures, and accessibility to optimize routing.

7. Search Functionality:

- Integrate a search feature allowing users to search for specific buildings, facilities, or landmarks within the campus.
- Implement autocomplete suggestions for efficient and accurate search results.

8. Favorite Locations Management:

- Provide users with the ability to save their favorite locations or frequently visited spots within the campus.
- Enable users to easily access and manage their saved locations for quick navigation.

9. User Feedback Integration:

- - Implement a feedback mechanism for users to provide input on the application's usability, accuracy, and features.
- - Regularly review and incorporate user feedback to improve the application's performance and user satisfaction.

CAMPUS NAVIGATION SYSTEM

10. Additional Features:

- **Active Users Location:** Provide administration with access to real-time location data of active users for security and monitoring purposes.
- **Location Information:**Provide detailed information about selected locations, including descriptions, images, and coordinates.

Software & Hardware Requirements:~

Hardware Requirements:

1.Server Infrastructure:

Sufficient computational resources to host the application backend, database, and APIs.

- Scalable infrastructure to handle varying loads during peak usage times.
- Reliable network connectivity to ensure uninterrupted service availability.

2. Client Devices:

Desktop computers, laptops, smartphones, and tablets for accessing the campus navigation system.

- Compatible web browsers or mobile applications for accessing the system interface.

3. GPS-enabled Devices:

Smartphones or tablets equipped with GPS capabilities for accurate location detection.

- GPS receivers with high accuracy for use in specific scenarios where precise location information is required.

Software Requirements:

1. Backend Technologies:

- **Programming Languages:**JavaScript (Node.js) for backend development.
- **Database:**MongoDB for storing map data, user information, and preferences.
- **Geolocation Services:** Integration with Geolocation APIs (e.g., Google Maps Geolocation API) for location detection and leaflet API.

2. Frontend Technologies:

- **Web Development:** HTML,CSS, JAVASCRIPT,REACTJS, BOOTSTRAP for building responsive web interfaces.

3. Mapping Services:

- Integration with mapping services such as Google Maps, Leaflet.js, or Mapbox for displaying campus maps and routing information.
- Access to mapping APIs for retrieving map data, displaying markers, and generating directions.

4. **User Interface Design Tools:**
   - Design tools like Figma for creating wireframes, maps,Blueprints, and user interface designs.

5. **Development Tools:**
   - Integrated Development Environments (IDE) such as Visual Studio Code for code editing and debugging.
   - Version Control Systems (Git) for collaborative development and code management.

6. **Security Measures:**
   - Encryption protocols (HTTPS) for securing data transmission between clients and servers.implementing twilio for otp generation.
   - Authentication and authorization mechanisms to control access to sensitive features and data using Firebase.

Test Case Specification:~

1. **Map Display:**
   - **Test Case 1:** Verify that the campus map loads correctly upon accessing the application.
   - **Test Case 2:** Check that key landmarks, buildings, and pathways are displayed accurately on the map.
   - **Test Case 3:** Ensure smooth navigation and zoom functionality within the map interface.

2. **User Location Detection:**
   - **Test Case 1:** Verify that the Geolocation API accurately detects the user's current location.
   - **Test Case 2:** Check for compatibility across different devices and browsers.

3. **Destination Selection:**
   - **Test Case 1:** Verify that users can select their destination by clicking on the map.
   - **Test Case 2:** Check that users can also choose from a list of predefined destinations.
   - **Test Case 3:** Ensure the selected destination is accurately reflected in the routing algorithm.

CAMPUS NAVIGATION SYSTEM

4. Step-by-Step Directions:

- **Test Case 1:** Verify that users receive accurate step-by-step directions to their destination.
- **Test Case 2:** Check that landmarks or key points are included in the directions for easy navigation.
- **Test Case 3:** Ensure the directions are updated in real-time to account for any changes in route or conditions.

5. Search Functionality:

- **Test Case 1:** Verify that users can search for specific buildings, facilities, or landmarks within the campus.
- **Test Case 2:** Ensure users can get directions to the searched location from their current position.

6. Save Favorite Locations:

- **Test Case 1:** Verify that users can save their favorite locations within the campus.
- **Test Case 2:** Check that saved locations are accessible for quick access in the future.
- **Test Case 3:** Ensure users can manage and delete saved locations as needed.

7. Active Users Location:

- **Test Case 1:** Verify that the administration can view the real-time location of active users.
- **Test Case 2**: Ensure the location tracking system operates efficiently and accurately.

8. Location Info:

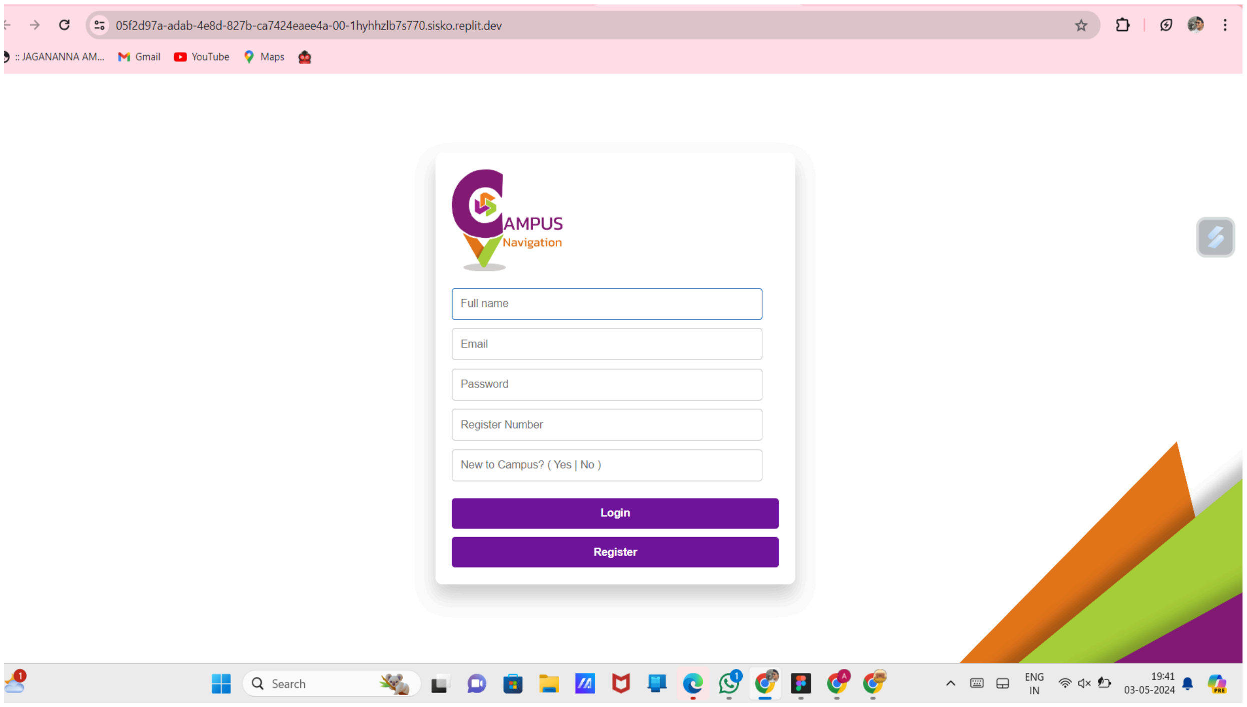
- **Test Case 1:** Verify that users can access detailed information about selected locations.
- **Test Case 2:** Check that location info includes relevant details such as descriptions, images, and coordinates.
- **Test Case 3:** Ensure the location info interface is user-friendly and accessible.

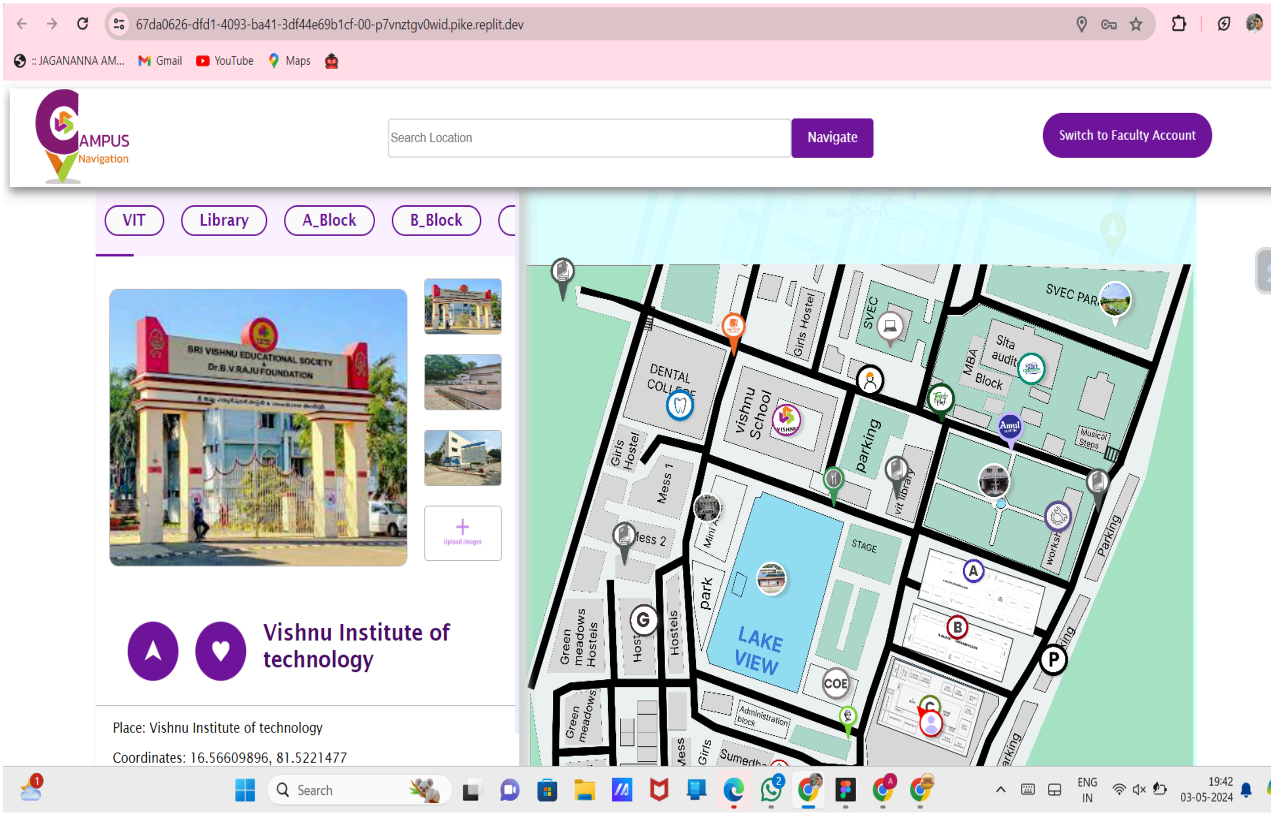
9. Feedback:

- **Test Case 1:** Verify that users can provide feedback on the application.
- **Test Case 2:** Check feedback is collected and stored for review by the team.
- **Test Case 3:** Ensure the application is updated based on the feedback received to improve user experience.

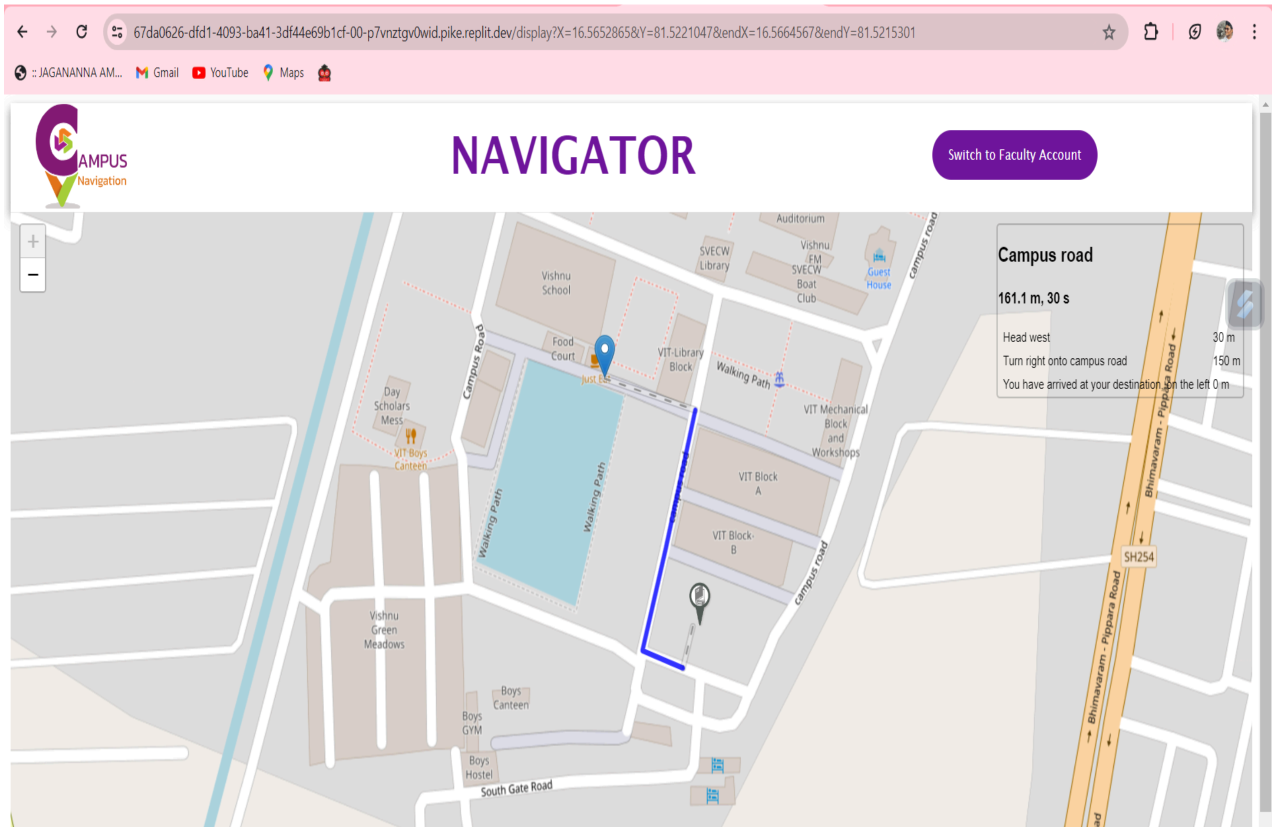
CAMPUS NAVIGATION SYSTEM

Output: YouTube link- https://www.youtube.com/watch?v=XB2rMsZYjj0


05f2d97a-adab-4e8d-827b-ca7424eaee4a-00-1hyhhzlb7s770.sisko.replit.dev
:: JAGANANNA AM... Gmail YouTube Maps
AMPUS
Navigation
Full name
Email
Password
Register Number
New to Campus? (Yes | No)
Login
Register
Search
ENG IN
19:41
03-05-2024



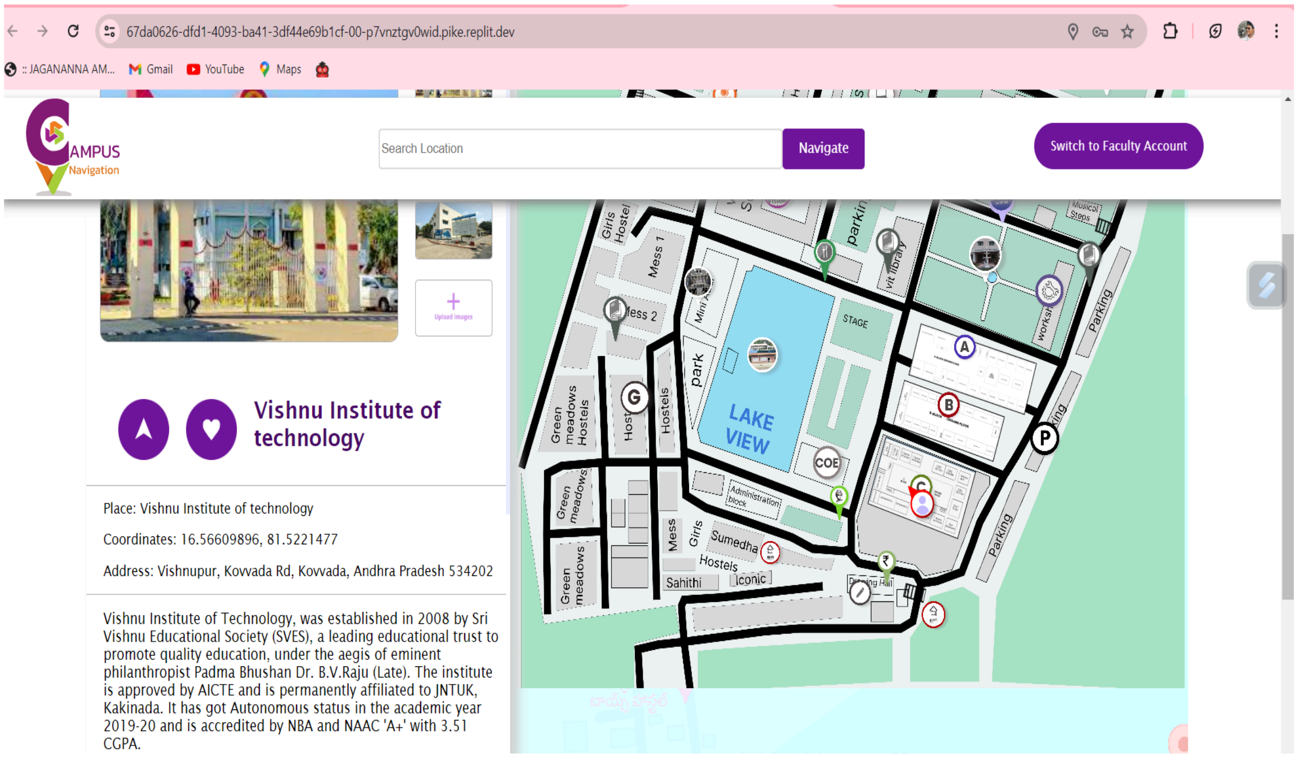
CAMPUS NAVIGATION SYSTEM


67da0626-dfd1-4093-ba41-3df44e69b1cf-00-p7vnztgv0wid.pike.replit.dev/display?X=16.5652865&Y=81.5221047&endX=16.5664567&endY=81.5215301
:: JAGANANNA AM... Gmail YouTube Maps
CAMPUS Navigation
NAVIGATOR
Switch to Faculty Account
Vishnu School
Food Court
VIT-Library Block
Walking Path
VIT Mechanical Block and Workshops
VIT Block A
VIT Block-B
Vishnu Green Meadows
Boys Canteen
Boys GYM
Boys Hostel
South Gate Road
SH254
Bhimavaram - Pippara Road

Campus road

161.1 m, 30 s

Head west	30 m
Turn right onto campus road	150 m
You have arrived at your destination, on the left	0 m



Vishnu Institute of technology

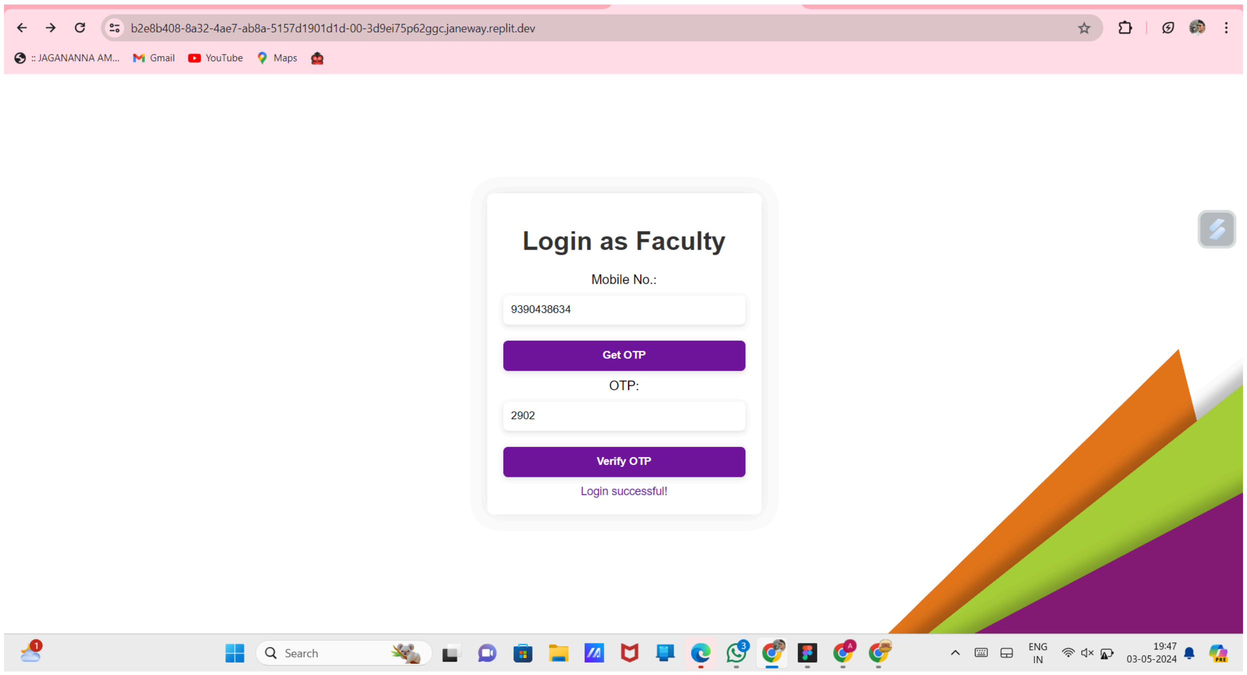
Place: Vishnu Institute of technology

Coordinates: 16.56609896, 81.5221477

Address: Vishnupur, Kovvada Rd, Kovvada, Andhra Pradesh 534202

Vishnu Institute of Technology, was established in 2008 by Sri Vishnu Educational Society (SVES), a leading educational trust to promote quality education, under the aegis of eminent philanthropist Padma Bhushan Dr. B.V.Raju (Late). The institute is approved by AICTE and is permanently affiliated to JNTUK, Kakinada. It has got Autonomous status in the academic year 2019-20 and is accredited by NBA and NAAC 'A+' with 3.51 CGPA.

CAMPUS NAVIGATION SYSTEM


b2e8b408-8a32-4ae7-ab8a-5157d1901d1d-00-3d9ei75p62ggc.janeway.replit.dev

Login as Faculty

Mobile No.:

9390438634

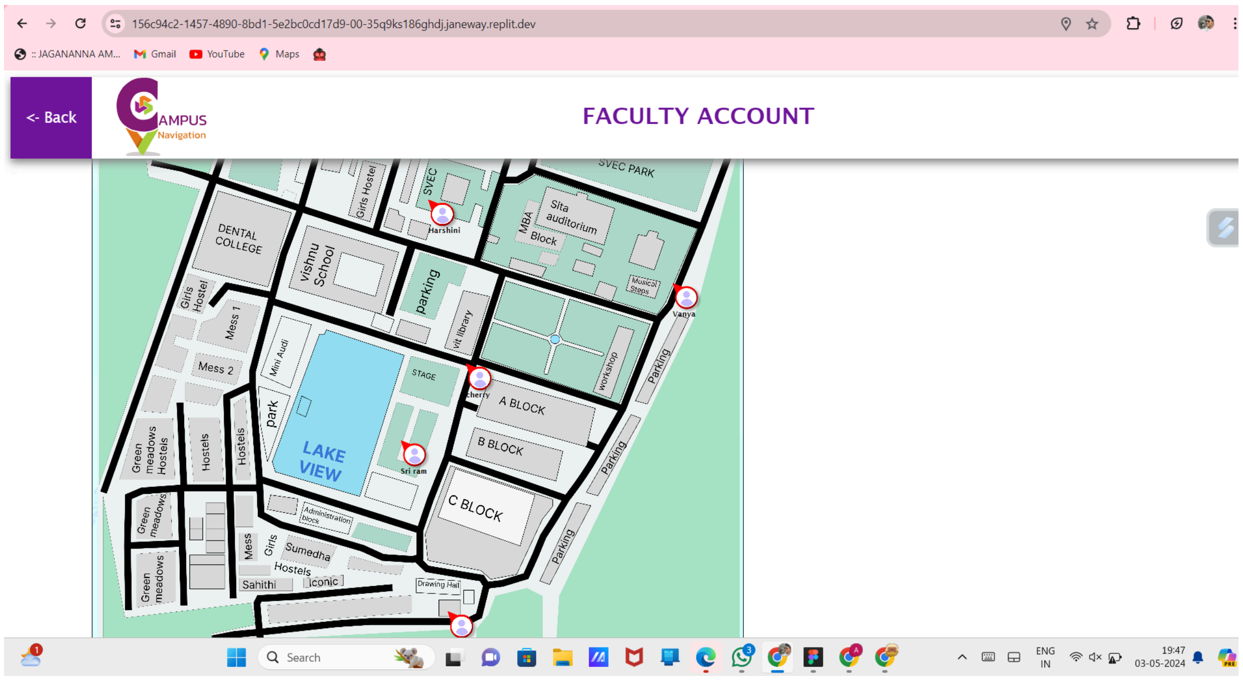
Get OTP

OTP:

2902

Verify OTP

Login successful!

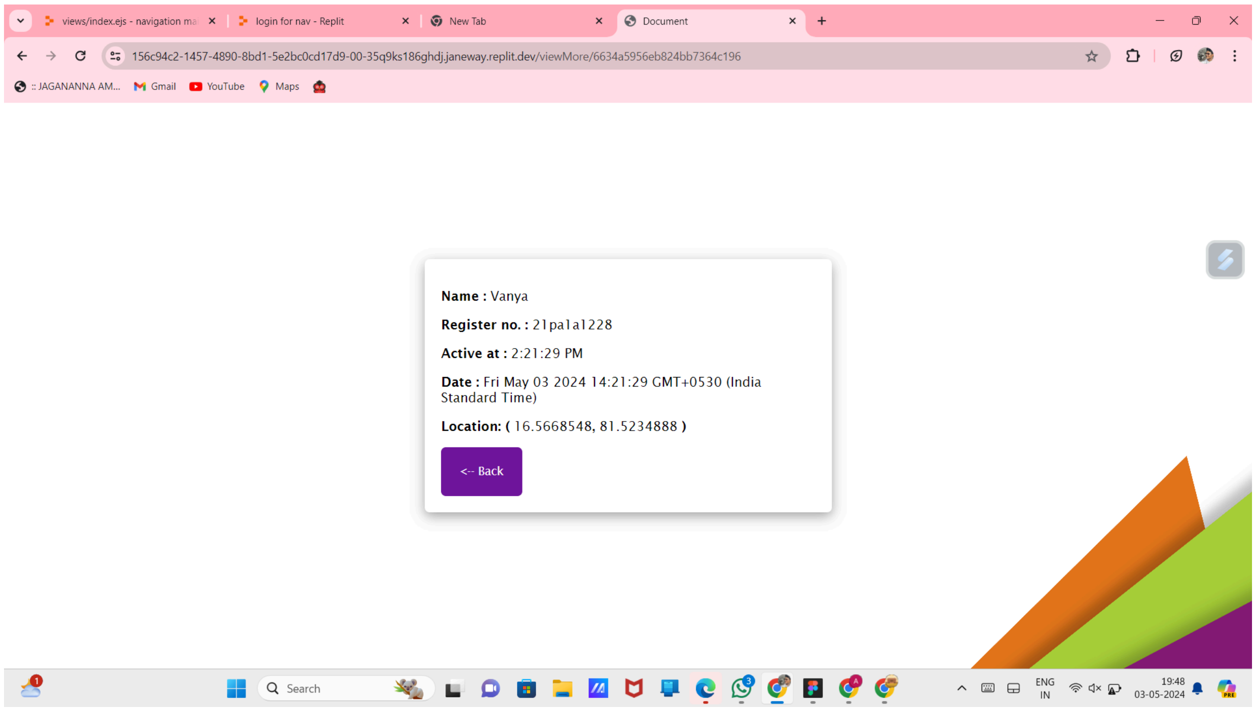

156c94c2-1457-4890-8bd1-5e2bc0cd17d9-00-35q9ks186ghdj.janeway.replit.dev

<- Back

CAMPUS Navigation

FACULTY ACCOUNT

CAMPUS NAVIGATION SYSTEM


views/index.ejs - navigation ma
login for nav - Replit
New Tab
Document
156c94c2-1457-4890-8bd1-5e2bc0cd17d9-00-35q9ks186ghdj.janeway.replit.dev/viewMore/6634a5956eb824bb7364c196
:: JAGANANNA AM...
Gmail
YouTube
Maps

Name : Vanya

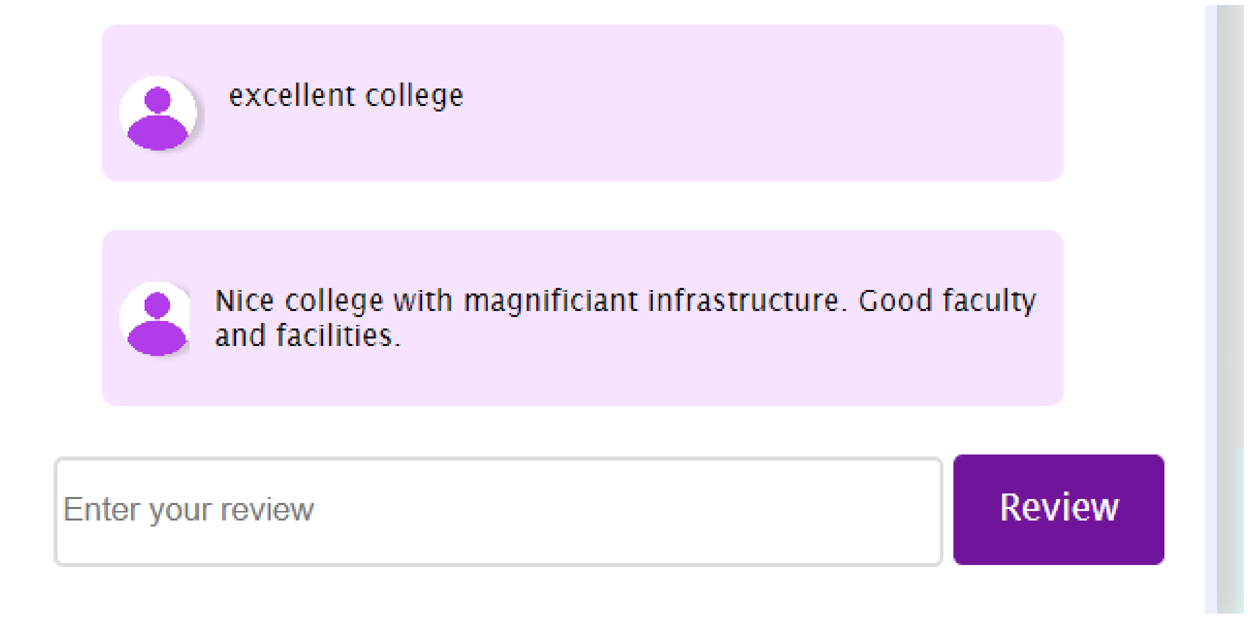
Register no. : 21pa1a1228

Active at : 2:21:29 PM

Date : Fri May 03 2024 14:21:29 GMT+0530 (India Standard Time)

Location: (16.5668548, 81.5234888)

<-- Back
Search
ENG
IN
19:48
03-05-2024

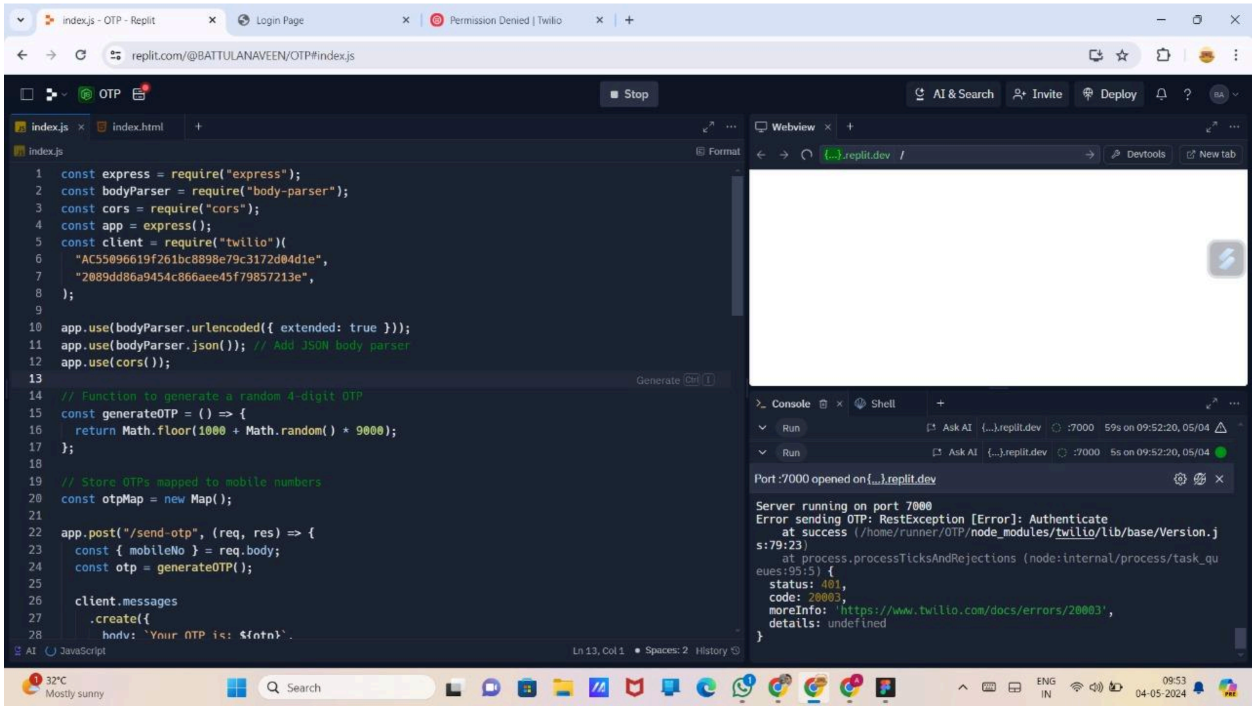


excellent college

Nice college with magnificiant infrastructure. Good faculty and facilities.

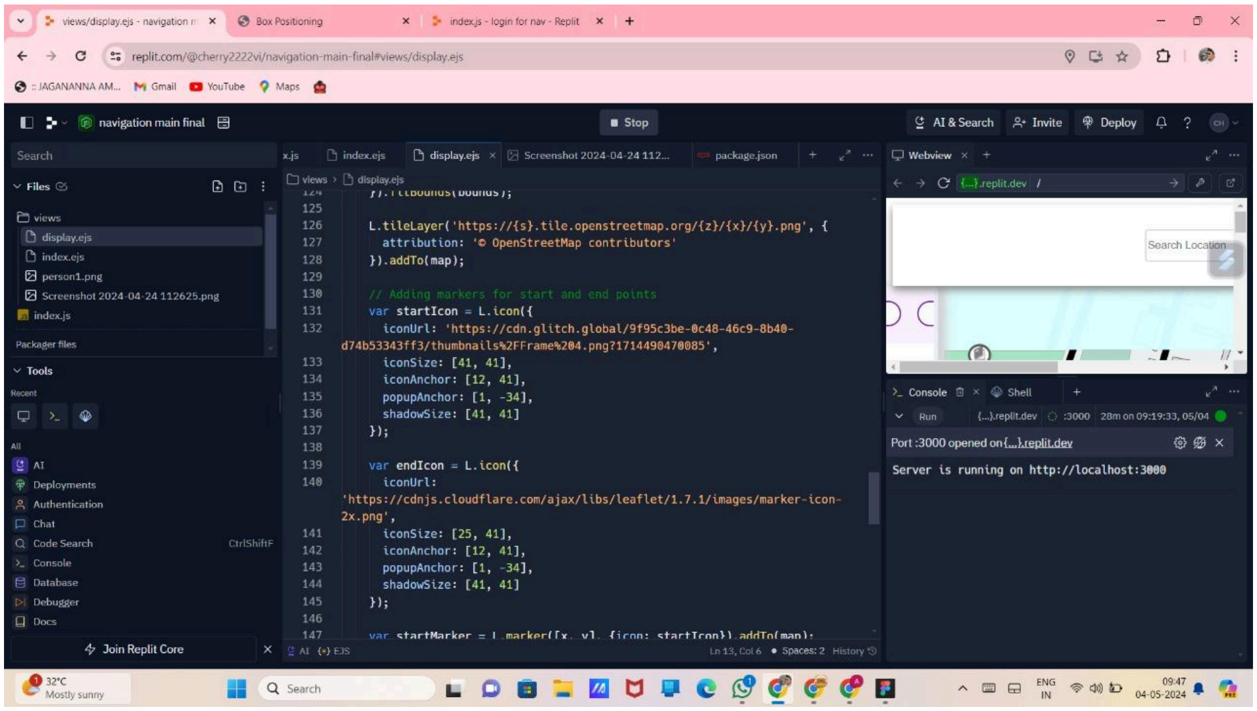
CAMPUS NAVIGATION SYSTEM

Code: **GitHub link- https://github.com/vanya7989/nevigation**



```
const express = require("express");
const bodyParser = require("body-parser");
const cors = require("cors");
const app = express();
const client = require("twilio")(
  "AC55096619f261bc8898e79c3172d04d1e",
  "2089dd86a9454c866aee45f79857213e",
);

app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json()); // Add JSON body parser
app.use(cors());

// Function to generate a random 4-digit OTP
const generateOTP = () => {
  return Math.floor(1000 + Math.random() * 9000);
};

// Store OTPs mapped to mobile numbers
const otpMap = new Map();

app.post("/send-otp", (req, res) => {
  const { mobileNo } = req.body;
  const otp = generateOTP();

  client.messages
    .create({
      body: `Your OTP is: ${otp}`,
```

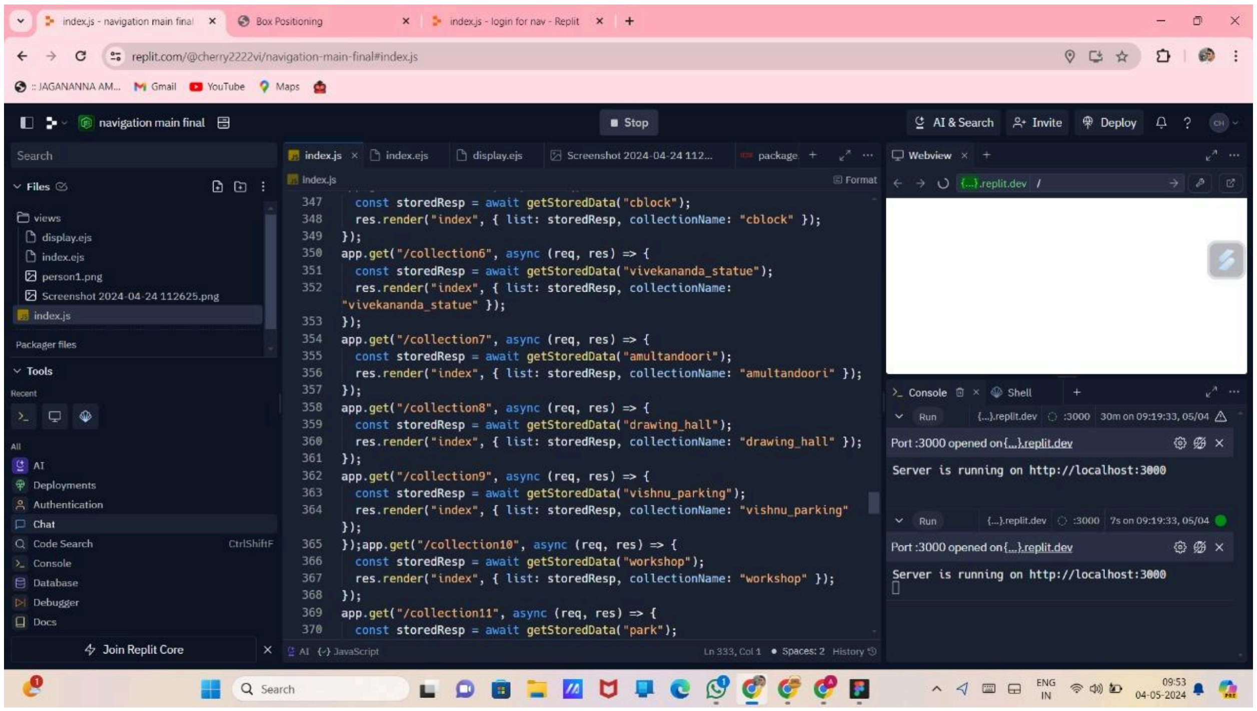
```
Port :7000 opened on {...}.replit.dev

Server running on port 7000
Error sending OTP: RestException [Error]: Authenticate
    at success (/home/runner/OTP/node_modules/twilio/lib/base/Version.j
s:79:23)
    at process.processTicksAndRejections (node:internal/process/task_qu
eues:95:5) {
  status: 401,
  code: 20003,
  moreInfo: 'https://www.twilio.com/docs/errors/20003',
  details: undefined
}
```



```
    }).fitBounds(bounds);

    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
      attribution: '© OpenStreetMap contributors'
    }).addTo(map);

    // Adding markers for start and end points
    var startIcon = L.icon({
      iconUrl: 'https://cdn.glitch.global/9f95c3be-0c48-46c9-8b40-d74b53343ff3/thumbnails%2FFrame%204.png?1714490470085',
      iconSize: [41, 41],
      iconAnchor: [12, 41],
      popupAnchor: [1, -34],
      shadowSize: [41, 41]
    });

    var endIcon = L.icon({
      iconUrl: 'https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.7.1/images/marker-icon-2x.png',
      iconSize: [25, 41],
      iconAnchor: [12, 41],
      popupAnchor: [1, -34],
      shadowSize: [41, 41]
    });

    var startMarker = L.marker([x, y], {icon: startIcon}).addTo(map);
```

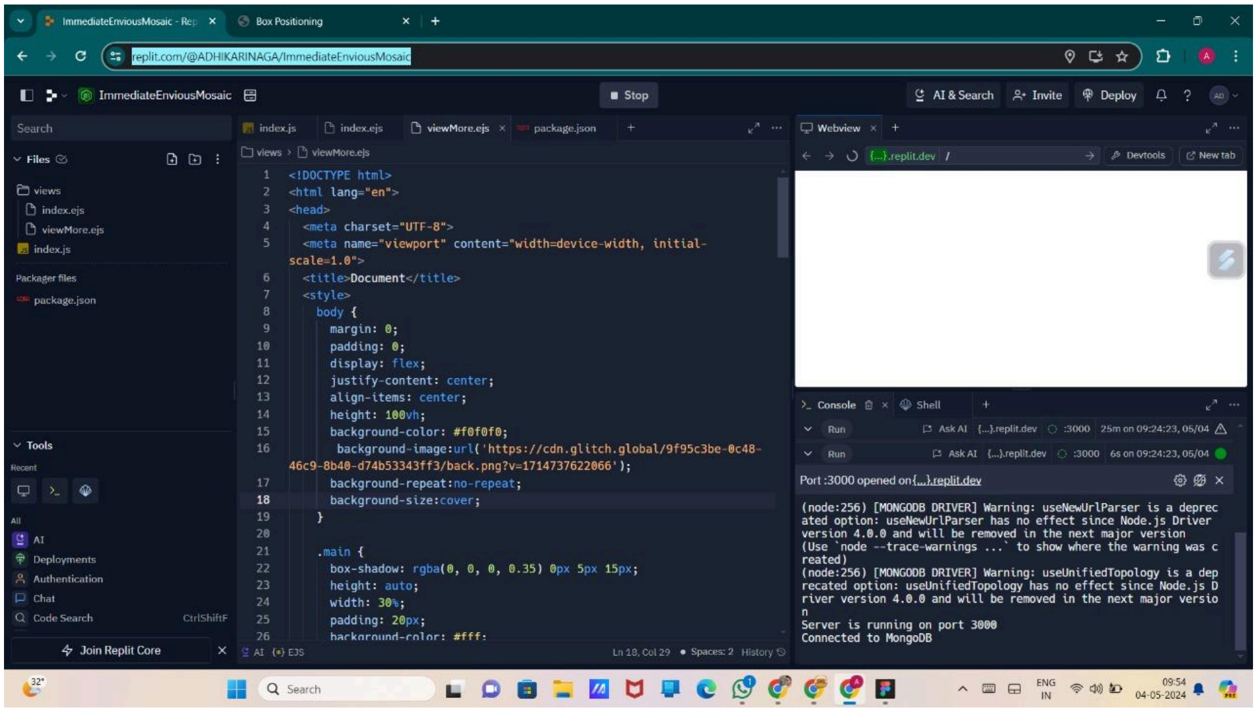
```
Port :3000 opened on {...}.replit.dev

Server is running on http://localhost:3000
```

CAMPUS NAVIGATION SYSTEM



```javascript
  const storedResp = await getStoredData("cblock");
  res.render("index", { list: storedResp, collectionName: "cblock" });
});
app.get("/collection6", async (req, res) => {
  const storedResp = await getStoredData("vivekananda_statue");
  res.render("index", { list: storedResp, collectionName:
"vivekananda_statue" });
});
app.get("/collection7", async (req, res) => {
  const storedResp = await getStoredData("amultandoori");
  res.render("index", { list: storedResp, collectionName: "amultandoori" });
});
app.get("/collection8", async (req, res) => {
  const storedResp = await getStoredData("drawing_hall");
  res.render("index", { list: storedResp, collectionName: "drawing_hall" });
});
app.get("/collection9", async (req, res) => {
  const storedResp = await getStoredData("vishnu_parking");
  res.render("index", { list: storedResp, collectionName: "vishnu_parking"
});
});app.get("/collection10", async (req, res) => {
  const storedResp = await getStoredData("workshop");
  res.render("index", { list: storedResp, collectionName: "workshop" });
});
app.get("/collection11", async (req, res) => {
  const storedResp = await getStoredData("park");
```

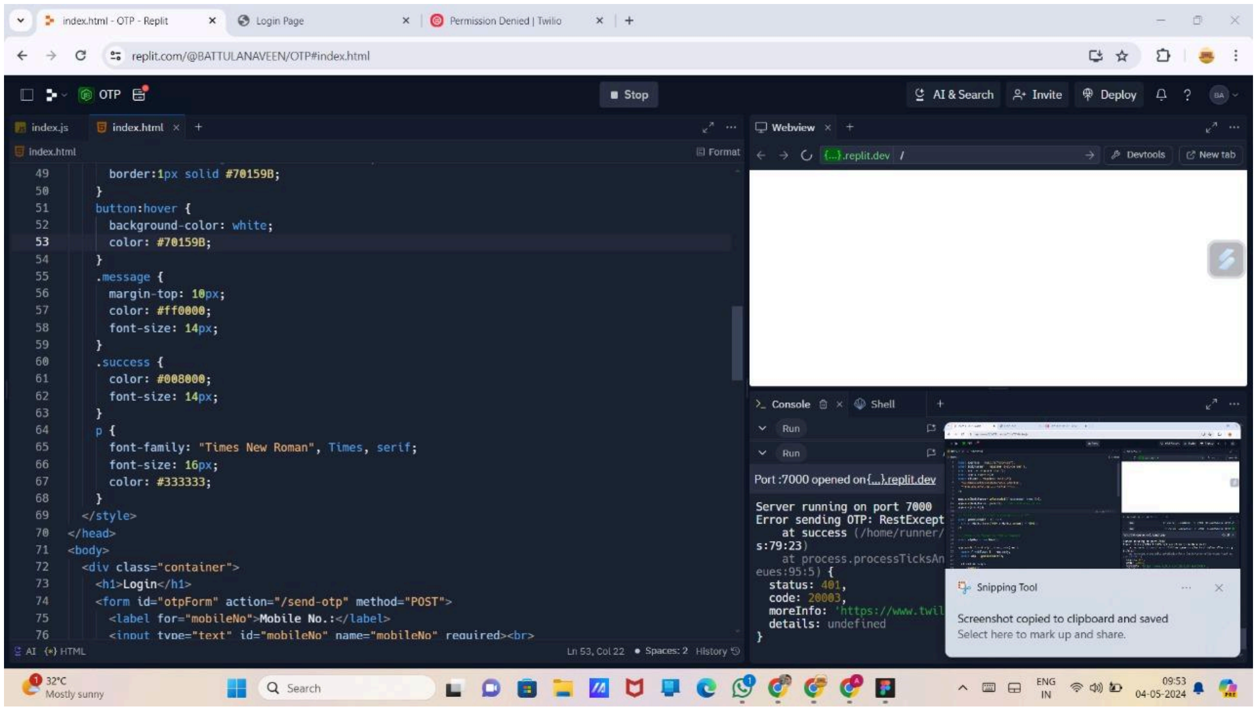
```
Port :3000 opened on {...}.replit.dev
Server is running on http://localhost:3000

Port :3000 opened on {...}.replit.dev
Server is running on http://localhost:3000
```



```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
  <style>
    body {
      margin: 0;
      padding: 0;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      background-color: #f0f0f0;
       background-image:url('https://cdn.glitch.global/9f95c3be-0c48-
46c9-8b40-d74b53343ff3/back.png?v=1714737622066');
      background-repeat:no-repeat;
      background-size:cover;
    }

    .main {
      box-shadow: rgba(0, 0, 0, 0.35) 0px 5px 15px;
      height: auto;
      width: 30%;
      padding: 20px;
      background-color: #fff;
```

```
Port :3000 opened on {...}.replit.dev
(node:256) [MONGODB DRIVER] Warning: useNewUrlParser is a deprec
ated option: useNewUrlParser has no effect since Node.js Driver
version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was c
reated)
(node:256) [MONGODB DRIVER] Warning: useUnifiedTopology is a dep
recated option: useUnifiedTopology has no effect since Node.js D
river version 4.0.0 and will be removed in the next major versio
n
Server is running on port 3000
Connected to MongoDB
```

CAMPUS NAVIGATION SYSTEM



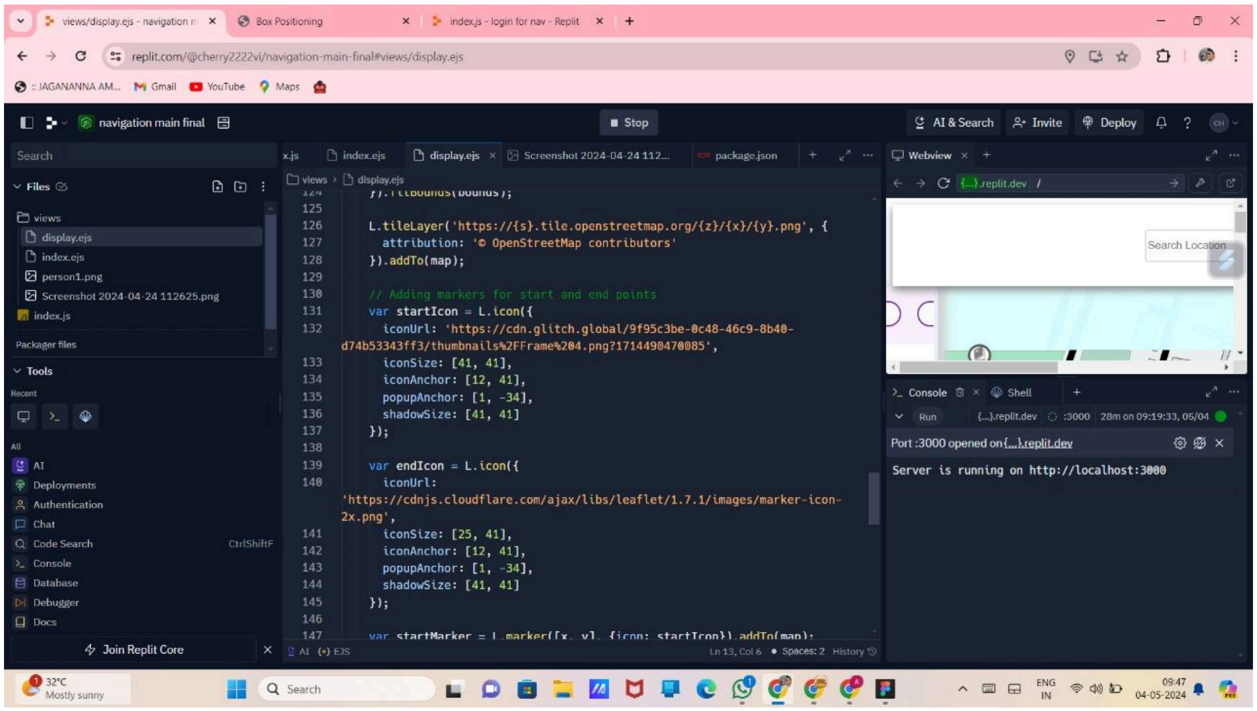
replit.com/@BATTULANAVEEN/OTP#index.html

```
      border:1px solid #70159B;
    }
    button:hover {
      background-color: white;
      color: #70159B;
    }
    .message {
      margin-top: 10px;
      color: #ff0000;
      font-size: 14px;
    }
    .success {
      color: #008000;
      font-size: 14px;
    }
    p {
      font-family: "Times New Roman", Times, serif;
      font-size: 16px;
      color: #333333;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>Login</h1>
    <form id="otpForm" action="/send-otp" method="POST">
      <label for="mobileNo">Mobile No.:</label>
      <input type="text" id="mobileNo" name="mobileNo" required><br>
```

Port :7000 opened on {...}.replit.dev

```
Server running on port 7000
Error sending OTP: RestExcept
    at success (/home/runner/
s:79:23)
    at process.processTicksAn
eues:95:5) {
  status: 401,
  code: 20003,
  moreInfo: 'https://www.twil
  details: undefined
}
```

Snipping Tool
Screenshot copied to clipboard and saved
Select here to mark up and share.



replit.com/@cherry2222vi/navigation-main-final#views/display.ejs

```
    }).fitBounds(bounds);

    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
      attribution: '© OpenStreetMap contributors'
    }).addTo(map);

    // Adding markers for start and end points
    var startIcon = L.icon({
      iconUrl: 'https://cdn.glitch.global/9f95c3be-0c48-46c9-8b40-
d74b53343ff3/thumbnails%2FFrame%204.png?1714490470085',
      iconSize: [41, 41],
      iconAnchor: [12, 41],
      popupAnchor: [1, -34],
      shadowSize: [41, 41]
    });

    var endIcon = L.icon({
      iconUrl:
'https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.7.1/images/marker-icon-
2x.png',
      iconSize: [25, 41],
      iconAnchor: [12, 41],
      popupAnchor: [1, -34],
      shadowSize: [41, 41]
    });

    var startMarker = L.marker([x, y], {icon: startIcon}).addTo(map);
```

Port :3000 opened on {...}.replit.dev

```
Server is running on http://localhost:3000
```

CAMPUS NAVIGATION SYSTEM

Future Scope:~

Future Scope for Campus Navigation System

1. Indoor Navigation:

- Extend navigation capabilities to include indoor areas of campus buildings.
- Implement indoor mapping technologies such as Bluetooth beacons or Wi-Fi positioning systems for accurate indoor location tracking.

2. Integration with Smart Campus Infrastructure:

- Integrate with IoT devices and sensors deployed across the campus to provide real-time data on factors like occupancy levels, environmental conditions, and building accessibility.
- Utilize this data to optimize routing decisions and enhance the overall navigation experience.

3. Integration with Campus Events and Activities:

- Provide information on campus events, workshops, seminars, and other activities within the navigation system.
- Enable users to navigate to specific event locations and receive updates on event schedules and changes.

4. Accessibility Features:

- Enhance accessibility features to cater to users with disabilities or special needs.
- Provide alternative navigation options, such as audio instructions and tactile feedback, for users with visual impairments.

5. Social Integration:

- Implement social features that allow users to share their location, routes, and favorite spots with friends and classmates.
- Facilitate social interactions and collaboration through the navigation system, such as organizing group outings or study sessions.

References:~

- Google Maps
- College Blueprint
- https://ieeexplore.ieee.org/abstract/document/9071669
- https://iopscience.iop.org/article/10.1088/1742- 6596/1997/1/012038/pdf

Conclusion:~

A campus navigation project using HTML, CSS, JavaScript, Bootstrap, MongoDB, and the Geolocation API provides a valuable service to users by helping them navigate around a campus or large area efficiently. By implementing key features such as map display, route generation, and step-by-step directions, the application can enhance the user experience and make navigation easier for students, visitors and staff members.

- **GitHub link- https://github.com/vanya7989/nevigation**
- **YouTube link- https://www.youtube.com/watch?v=XB2rMsZYjj0**