

## **B9 A10 Type-01 Requirements**

### **Task Description**

You must create a tourism-focused website, but not just any tourism! We're concentrating on a specific kind.

---

### **Website Themes: (Tourism Management website)**

We've listed six various countries of different zones. Here **Southeast Asia, and Europe- zones are the categories and countries are the sub-categories**. Select one specific from the list below to start building your website. You cannot include properties from other categories, such as if you choose the Southeast Asia category on your website, you can not use other countries' names and tourist spots like Europe. For better understanding, we have added tourist spots for corresponding countries. You have to create data for these types of tourist spots.

#### **1. Southeast Asia**

- a. Bangladesh: Sundarban, Cox's Bazar, Rangamati, Bandarban, Saint Martin's Island
- b. Thailand: Bangkok, Chiang Mai, Ayutthaya, Phuket, Phi Phi Islands
- c. Indonesia: Bali, Borobudur Temple, Komodo National Park, Raja Ampat Islands, Yogyakarta
- d. Malaysia: Kuala Lumpur, Langkawi, Penang, Cameron Highlands, Taman Negara National Park
- e. Vietnam: Ha Long Bay, Ho Chi Minh City, Hoi An Ancient Town, Phong Nha Caves, Mekong Delta
- f. Cambodia: Angkor Wat, Siem Reap, Phnom Penh, Kep, Bokor National Park

#### **2. Europe**

- a. France
- b. Italy
- c. Spain
- d. England
- e. Netherlands
- f. Switzerland

### **3. America**

- a. United States of America
- b. Canada
- c. Brazil
- d. Argentina
- e. Mexico
- f. Peru

### **4. Africa**

- a. South Africa
- b. Tanzania
- c. Kenya
- d. Morocco
- e. Egypt
- f. Namibia

### **5. Middle East**

- a. Jordan
- b. Saudi Arabia
- c. United Arab Emirates
- d. Iran
- e. Qatar
- f. Kuwait

### **6. Central Asia**

- a. Uzbekistan
- b. Kyrgyzstan
- c. Kazakhstan
- d. Tajikistan
- e. Turkmenistan
- f. Mongolia

---

## **Main Requirements:**

### **Key Rules:**

- Include a minimum of 15 notable GitHub commits on the client side
- Include a minimum of 8 notable GitHub commits on the server side

- Add a meaningful readme.md file with the name of your website and a live site URL. Include a minimum of five bullet points to feature your website. Write your selected category's name here.
  - Make it responsive for all devices. You need to make it responsive for mobile, tablet and desktop views.
  - After reloading the page of a private route, the user should not be redirected to the login page.
  - Use the Environment variable to hide the Firebase config keys and MongoDB credentials.
  - Don't use any Lorem ipsum text; you can not use the default alert to show any error or success message.
- 

1. **Ensure that your design is creative and unique.** Choose the type of website you wish to create first. Next, look for ideas for the design on the internet or by visiting sites such as ThemeForest. But keep in mind that your idea for a website should look different from any past projects of yours or any of the samples from our modules or conceptual sessions.

🔗 You can also look for free resources on blogs to help with your website.  
<https://bootcamp.uxdesign.cc/free-images-and-resources-collection-for-website-c77f2fc46ce5>

## 2. **Navbar:** The navbar will contain the

- Website name/logo,
- Home,
- All Tourists Spot,
- Add Tourists Spot(**It will be a private/protected route.** ),
- My List (**It will be a private/protected route.** ),
- “Login”, and “Register”.

- a. The “Login” and “Register” button is conditional, if the user is not logged in it will show the “Login” and “Register” Button. If a user clicks on “Login” it will redirect to the login page. If a user clicks on “Register” it will redirect to the Register page.

- b. But If the user is logged in here you will show the user photoURL, when you hover over the image it will show the displayName. And it will show the “Log out” button.

**3. Login Page:** When you click the login button on the navbar it redirects to the login page. You have to use a password and email-based authentication to log in. The login page will have-

- a. Email
- b. Password
- c. Google login,
- d. GitHub/Facebook/Twitter login - implement any of one
- e. A link that will redirect to the Register page

🔒 Here the email and password should match with the registered email and password. If it doesn't match, show an error message. You can show an error by using toast/sweet alert if you want.

**4. Register Page:** You have to use a password and email-based authentication to register. The Register page will have the following -

- a. Name
- b. Email
- c. photoURL
- d. password
- e. A Link that will redirect to the login page

★ For password verification you need to follow this -

- Must have an Uppercase letter in the password
- Must have a Lowercase letter in the password
- Length must be at least 6 character

★ If any of this isn't fulfilled it will show an error message /toast

★ After successful login or Register you need to show toast/sweet alert

🎯 Don't implement email verification or forget password method as it will inconvenience the examiner. If you want, you can add these after receiving the assignment result.

5. **Home Page:** The home page will have the following sections- Navbar, Banner/Slider, Tourists Spots, Countries Section, 2 Extra Sections, and Footer.

🎯 Make sure to keep the navbar and footer on all the pages except the 404 page.

6. **Banner:** Add a slider (you can use any type of slider/carousel) with a minimum of 3 slides and meaningful information

7. **Tourists Spots Section:** Here you will show at least 6 cards of tourists spot you have added on database. You can show information as you want but make sure that the UI looks good. There will be a "View Details" Button which will redirect you to the "View Details page".

8. **Extra Section:** Add 2 relevant and meaningful extra sections which are mentioned above on the Home page.

9. **Add Tourists Spot Page:** Create an Add Tourists Spot page where there will be a form for the user to add a Tourists Spot. The form will have:

- a. image ( use image URL)
- b. tourists\_spot\_name
- c. country\_Name
- d. location
- e. short description
- f. average\_cost
- g. seasonality - like summer, winter
- h. travel\_time => like- 7 days
- i. totaVisitorsPerYear => like- 10000

- j. User Email
- k. User Name
- l. "Add" button

**This will be a private/protected route.**

When you fill in the data and submit the "Add" button, these data will be stored in your database and you will show a success message through toast/sweet alert.

**10. All Tourists Spot page:** Create an All Tourists Spot page where you will see the tourist spots all the users have added to the database. You will get the data from the database and show it here on the card. Each card will have the following information:

- a. image
- b. tourists\_spot\_name
- c. average\_cost
- d. totaVisitorsPerYear
- e. travel\_time
- f. seasonality
- g. "View Details" button – will redirect to the "View Details Page"

- At the top of this page, there will be a dropdown menu where you will implement sort functionality based on "average\_cost". You can sort by ascending or descending or, it's up to you.

**11. View Details Page: The Tourists Spot detail route will be a private/protected route.** Please make sure that if the user is not logged in, the private route redirects to the login page. On this page, you will show all the information you have stored in the database.

**12. My List Page: It will be a private/protected route.** On this page, a user can see all the tourists spot he/she has added to the database. here a user can only see his/her added data, but he/she can not access other's data. Here you need to show the data in a tabular form. Each row of the table will contain these information-

- a. 3-4 information about the tourist spot
- b. "Update" Button
- c. "Delete" Button

**13. Update Page:** When a user clicks on the "Update Button" it will take the user to the Update page, where there will be a form for the user to update a Tourist Spot. The form will have the following:

- a. image ( use image URL)
- b. tourists\_spot\_name
- c. country\_Name
- d. location
- e. short description
- f. average\_cost
- g. seasonality - like summer, winter
- h. travel\_time => like- 7 days
- i. totaVisitorsPerYear => like- 10000
- j. "Update" button

**This will be a private/protected route.** When you fill in the data and submit the "Update" button, these data will be updated to the previous data in your database and you will show a success message through toast/sweet alert.

🎯🎯**(Optional):** If you don't want to create an Update page, you can also use a **modal** to update your data. For this when you click on the "Update" button it will open a modal but make sure you are logged in before updating the data.

🎯**For all the CRUD operations, show relevant toast/ notification/ sweet alert with a meaningful message**

**14. Delete Button-** If the user clicks the delete button, the Tourists' spot will be removed from the data. Before the delete, ask for a delete confirmation.

**15. Footer:** Create a reasonable and meaningful footer. (including website name, copyright, contact information, social media links)

**16. 404 page:** Add a 404 page/Not Found Page

17. Show a loading spinner when the data is in a loading state.

---

## Challenges Requirements:

1. **Countries:** In this section, you will show at least 6 countries' information for your selected category. The information is - image, Country name and a short description. (For this you need to create another collection in your database and the country names should match with the tourist spot data "country\_Name".)
2. On clicking a "Country Card" on the Home page it will redirect the user to a page where the user can see all the tourist spot cards of that specific country. Each tourist spot card will have:
  - a. tourists\_spot\_name
  - b. country\_Name
  - c. location
  - d. short description
  - e. average\_cost
  - f. seasonality
  - g. "View Details" button

This "View Details button" will redirect to the "View Details Page".

3. Implement a dark/light theme toggle for the home page.
4. Explore this package and implement at least 2-
  - a. [Lottie React](#)
  - b. [React-simple-typewriter](#)
  - c. [React Awesome reveal](#)
  - d. [React-tooltip](#)

### What to submit:

- Assignment Requirement Type
- Your client-side code GitHub repository
- Your server-side code GitHub repository
- Your live website link