{Help and Support}
{Site Web}
{Travel}

{Developed by :LALOUI Abdelbari}
{Email:bari.2004uae@gmail.com }
{Phone:0556 75 31 82}
{GTR-L2-2023.2024-MT:222231649903}
{Algeria}

{ *Introduction* }

Welcome to our travel booking platform, your gateway to unforgettable adventures around the globe. At our site, we strive to make your travel dreams a reality by offering a seamless booking experience for a wide range of tour packages and holiday deals. Whether you're seeking a relaxing beach getaway, an exhilarating city escape, or an immersive cultural journey, we have something for every type of traveler. With user-friendly navigation and a vast selection of destinations and activities, planning your next vacation has never been easier.

This report provides a detailed overview of the services offered by our travel website and its unique features. Our services cater to every aspect of your travel needs, from booking flights and accommodations to arranging guided tours and excursions. With our commitment to customer satisfaction and expertise in the travel industry, we ensure that every trip booked through our platform is an unforgettable experience. Join us as we embark on a journey of exploration, discovery, and unforgettable memories.

{ *The Problem* }

Navigating through the complexities of traditional travel agency websites or making phone calls to book your dream vacation package can be cumbersome and time-consuming. However, our solution simplifies this process by providing a user-friendly online platform. Gone are the days of visiting multiple offices or spending hours on the phone trying to find the perfect deal. With our website, you can explore various travel options, compare prices, and make bookings—all in one convenient location.

{ *Objectives* }

1. Provide a user-friendly platform for travelers to easily search, compare, and book various travel services such as flights, accommodations, tours, and activities within comprehensive tourism packages.

2. Offer a comprehensive database of destinations,, , and tour packages to cater to diverse traveler preferences and budgets.

3. Ensure secure online transactions and data privacy to instill trust and confidence among users.

4. Integrate real-time availability and pricing updates to enable instant booking confirmations and minimize booking errors within comprehensive tourism packages.

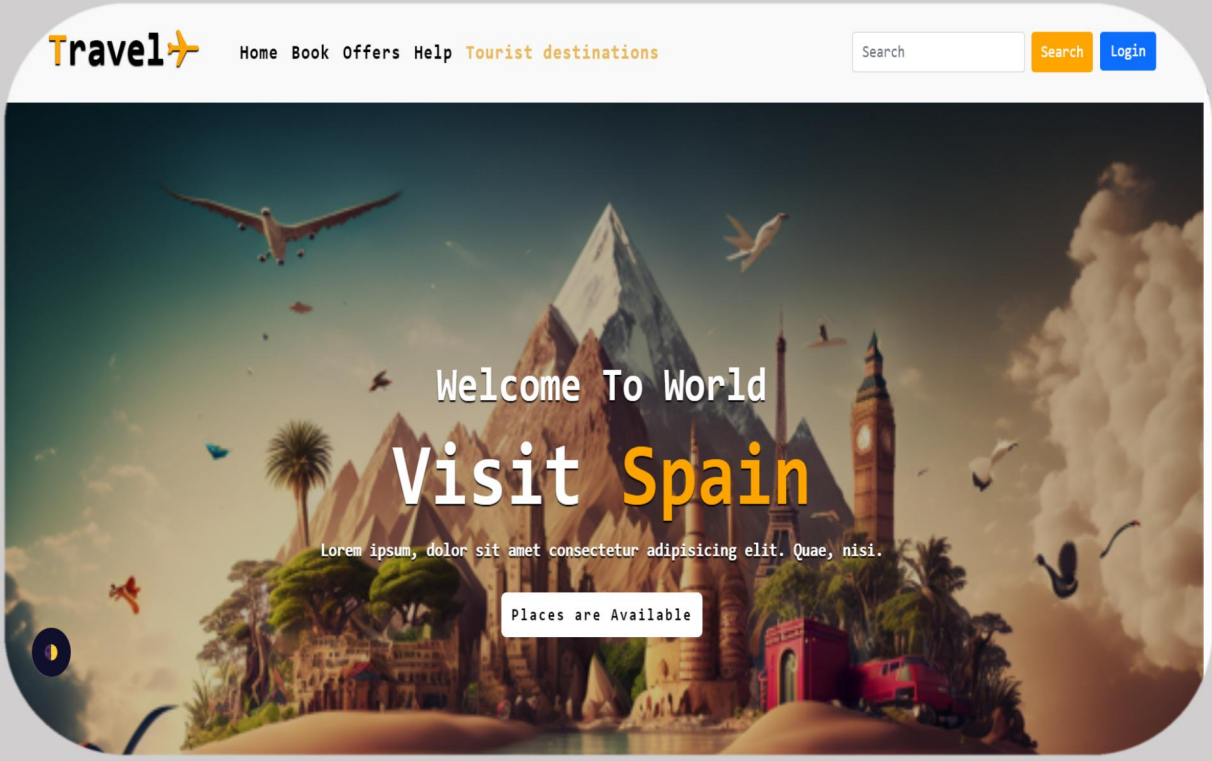
5. Implement responsive design to optimize the website for seamless access and functionality across multiple devices and screen sizes.

6. Offer customer support channels such as live chat, email, and phone assistance to address inquiries, resolve issues, and provide assistance throughout the booking process and travel journey within comprehensive tourism packages.

7. Continuously analyze user feedback, website performance metrics, and market trends to identify areas for improvement and innovation, ensuring ongoing relevance and competitiveness in the dynamic tourism industry.

8. Develop intuitive and efficient user interfaces for both administrators and users, ensuring smooth navigation, clear information presentation, and easy access to relevant features and functions within the comprehensive tourism packages.

{ Visitor-Interface}



(1)

Upon entering the website, a following page appears featuring a dynamic screen displaying some of the most popular high-demand destinations.

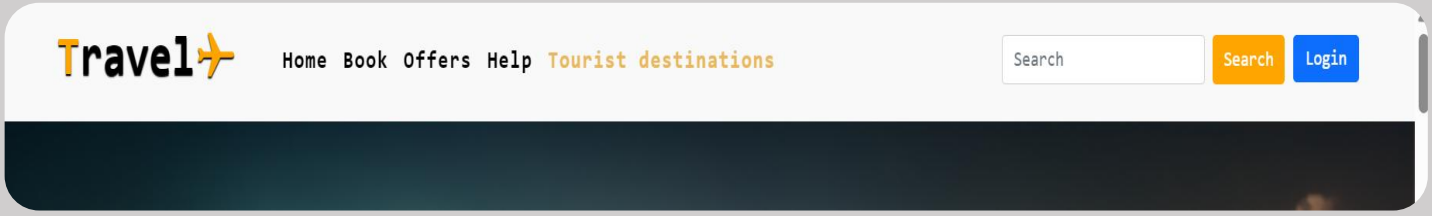
--> HTML-CSS:

```html
<nav class="navbar navbar-expand-lg" id="navbar">
    <div class="container">
      <div class="collapse navbar-collapse" id="mynavbar">
            <li class="nav-item">

              <a class="btn btn-primary" href="uase.html" >Login</a>


            </li>
            </form>
          </form>
        </div>
      </div>
    </nav>

<div class="home">
    <div class="content">
        <h5>Welcome To World</h5>
        <h1>Visit <span class="changecontent"></span></h1>
        <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Quae, nisi.</p>
        <a href="BookPlace1.html">Places are Available</a>
    </div>
```

```css
-item .nav-link{ ···
nav-item .nav-link:hover{ ···
navbar form button{ ···

home{ ···
home .content{ ···
home .content h5{ ···
home .content h1{ ···
changecontent::after{ ···
keyframes changetext{ ···
me .content p{ ···
```

HTML and CSS were used to divide the following section using class and section elements in order to style its appearance with CSS.

[Task list]



(2)

Website navigation bar allows seamless movement throughout the site with eas:

- Home: Takes you back to the main page.

- Book: Directs you to a suitable place for making reservations.
- Help: Contains step-by-step instructions and educational PDFs to assist in performing tasks on the website.
- Tourist destination: Displays a list of continents where the site offers tourist attractions
.
- Login: For creating an account on the websit.

(3)

--> HTML-CSS:

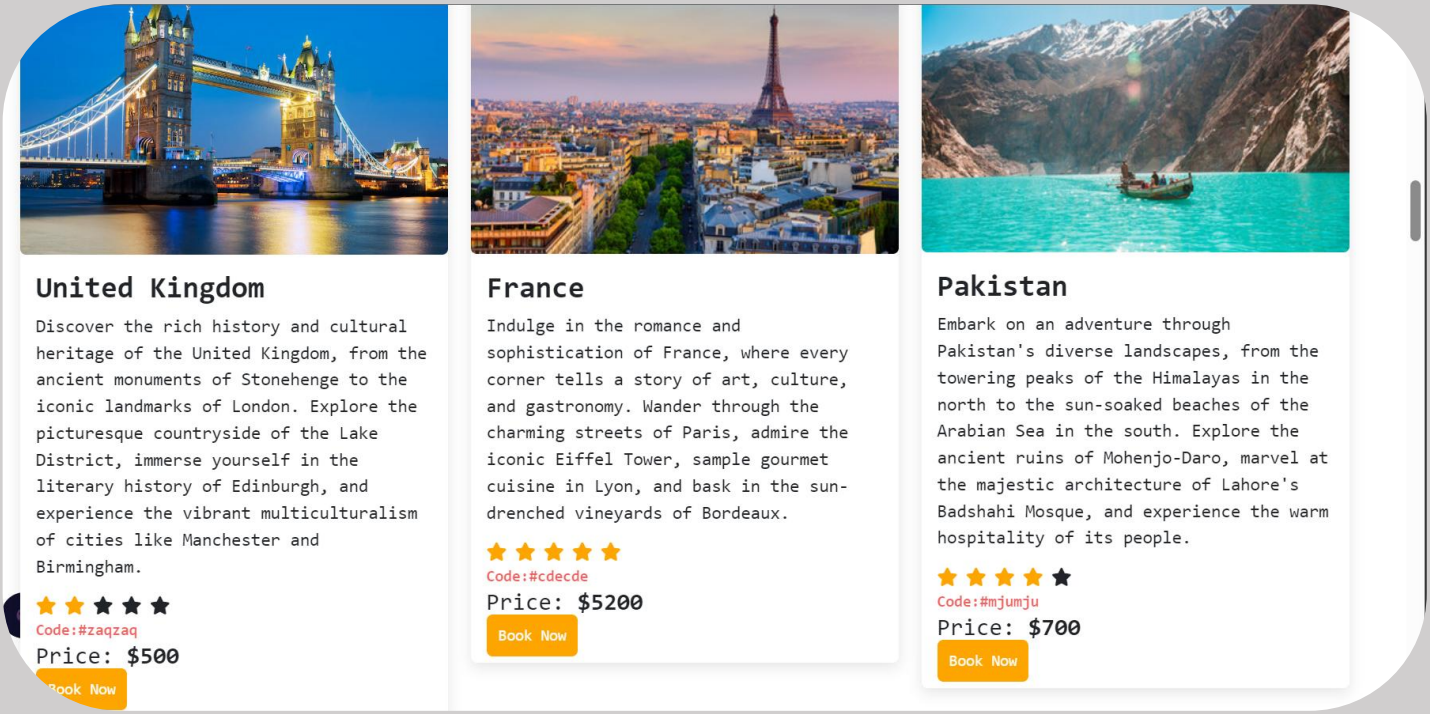
```
<body>
    <nav class="navbar navbar-expand-lg" id="navbar">
        <div class="container">
            </button>
            <div class="collapse navbar-collapse" id="mynavbar">
              <ul class="navbar-nav me-auto">
                <li class="nav-item">
                  <a class="nav-link active" href="b.html">Home</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="b.html#book">Book</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="Offers1.html">Offers</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="Help1.html">Help</a>
                </li>

                <li class="nav-item">
                  <a class="nav-link" href="#gallary"><span style="color: #eab95f;">Tourist destinations</span></a>
                  <ul class="g">
                    <li><a href="Africa1.html" style="color: antiquewhite;">Africa</a>
                    <li><a href="Asia1.html" style="color: antiquewhite;">Asia</a>
```

```
style>*{
logo{ ···
}
#logo span{ ···
}
.navbar-toggler span{ ···
}
.error-message {
        display: none;
        color: red;
    }
    .error-message2 { ···
    }
.navbar-nav{ ···
}
.nav-item .nav-link{ ···
}
.nav-item .nav-link:hover{ ···
}
#navbar form button{ ···
}

.home{ ···

OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS
```

Snipping Tool

il and ul Command were used to create a task list and enhance its appearance by placing it appropriately and applying suitable color and style using CSS.

Packages



(4)

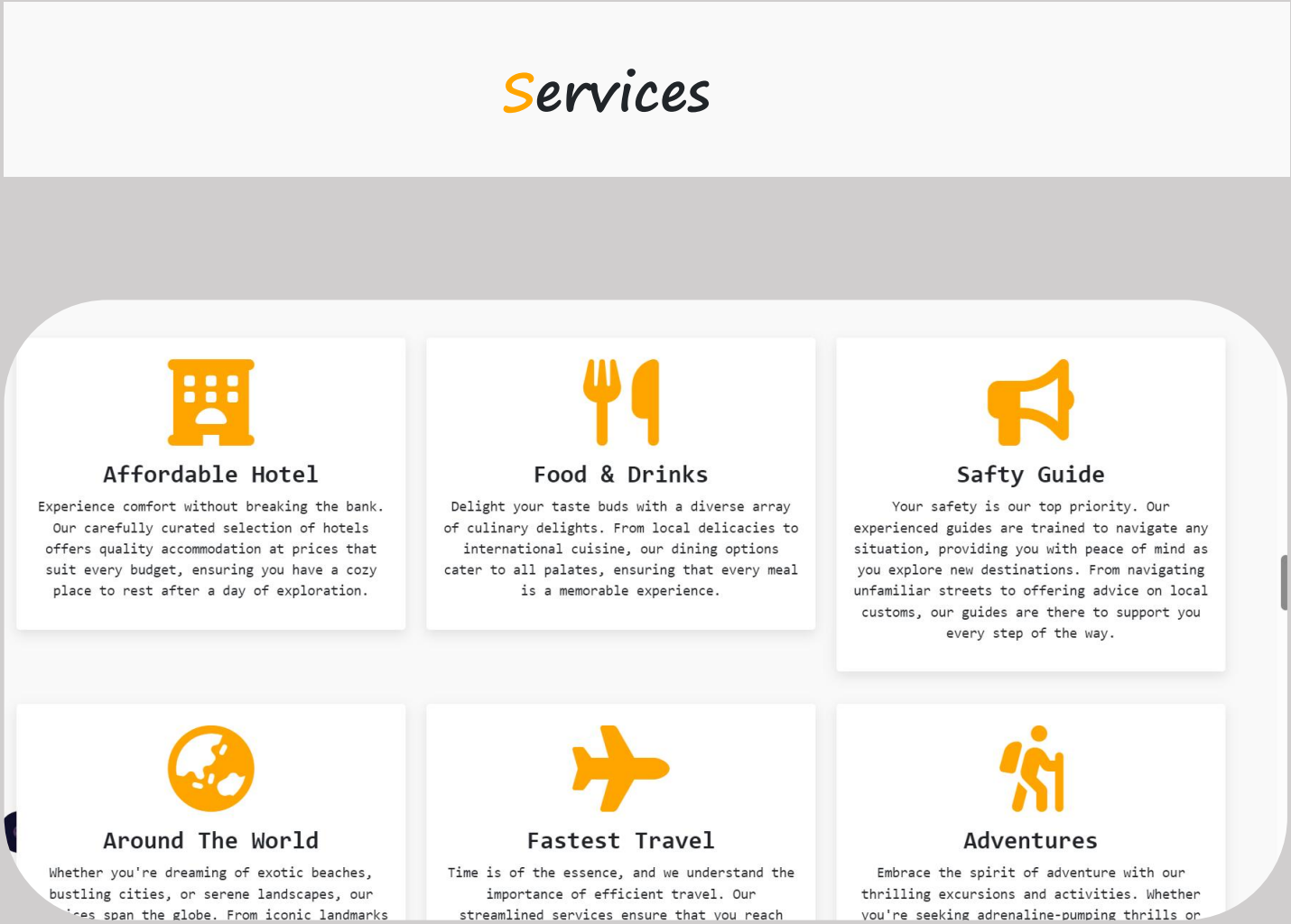
In another section of the home page, tourist packages are displayed, showcasing the price, location description, package price, and package symbol.

--> HTML-CSS:

```
ain-txt h1 span{ ···
}
.packages .card{ ···
}
.packages .card img{ ···
}
.packages .card .card-body{ ···
}
.packages .card .card-body h3{ ···
}
.packages .card .card-body p{ ···
}
.checked{ ···
}
.star i{ ···
}
.packages .card .card-body h6{ ···
}
packages .card .card-body a{ ···
```

```html
</section>
<section class="packages" id="packages">
  <div class="container">

    <div class="main-txt">
      <h1><span>P</span>ackages</h1>
    </div>

    <div class="row" style="margin-top: 30px;">

      <div class="col-md-4 py-3 py-md-0">

        <div class="card">
          <img src="./images/uk.png" alt="">
          <div class="card-body">
            <h3>United Kingdom</h3>
            <p>Discover the rich history and cultural heritage of the United Kingdom, from the ancient monuments of Stonehenge to the i
            <div class="star">
              <i class="fa-solid fa-star checked"></i>
              <i class="fa-solid fa-star checked"></i>
              <i class="fa-solid fa-star "></i>
              <i class="fa-solid fa-star "></i>
              <i class="fa-solid fa-star "></i>
```

Using div and classes, a specific part of the HTML has been given a name where it is styled using CSS.



The third section, which displays our unique services.

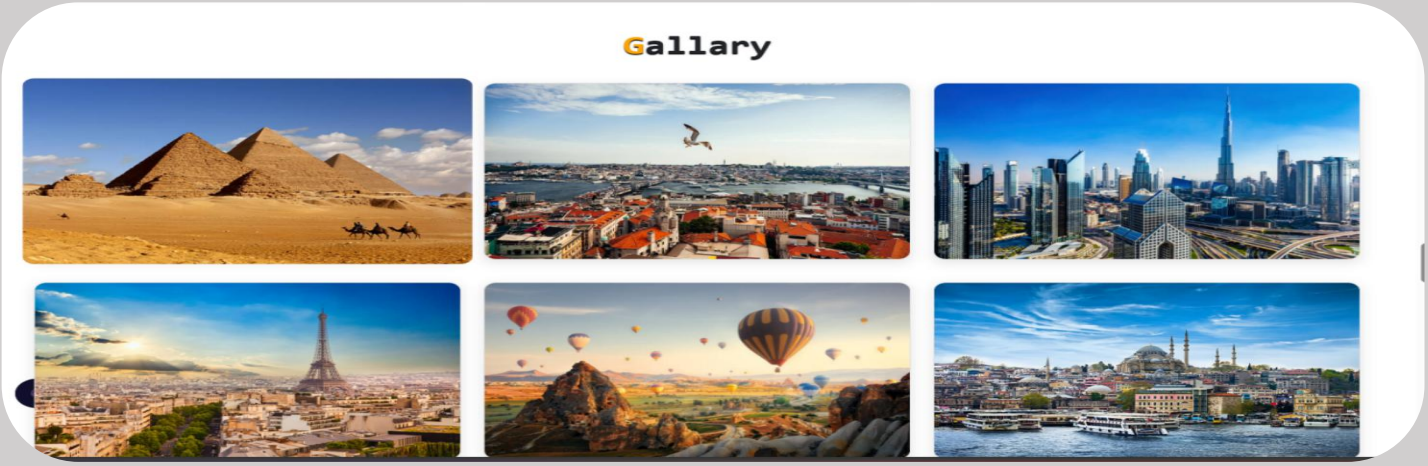
--> HTML-CSS:

```html
</section>

<section class="services" id="services">
 <div class="container">

   <div class="main-txt">
     <h1><span>S</span>ervices</h1>
   </div>

   <div class="row" style="margin-top: 30px;">

     <div class="col-md-4 py-3 py-md-0">

       <div class="card">
         <i class="fas fa-hotel"></i>
         <div class="card-body">
           <h3>Affordable Hotel</h3>
           <p>Experience comfort without breaking the bank. Our carefully curated selection of hotels offers quality accommodation at
         </div>
       </div>

     </div>
     <div class="col-md-4 py-3 py-md-0">
```

```css
.services{ ···
}
.services .card{ ···
}
.services .card i{ ···
}
.services .card .card-body h3{ ···
}
.services .card .card-body{ ···
}

.gallary{ ···
}
.gallary .card{ ···
}
.gallary .card img{ ···
}
.gallary .card img:hover{ ···
}

bout{ ···
```

Snipping Tool

Gallary

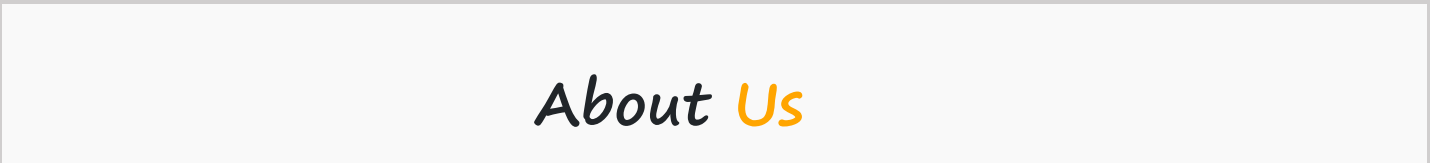


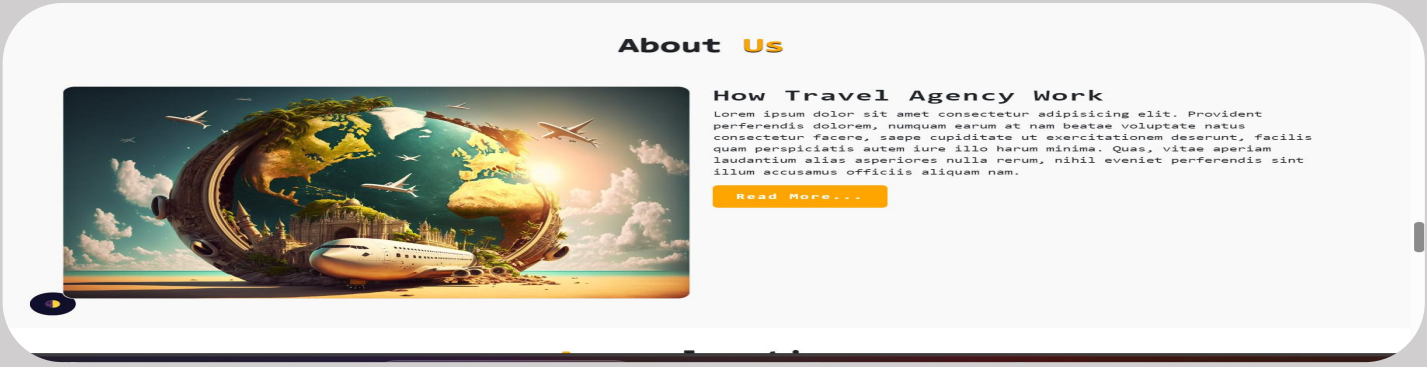
Section 4 displays a set of images captured by tourists who booked through us.

--> HTML-CSS:

```
> .gallary{ ···
  }
> .gallary .card{ ···
  }
> .gallary .card img{ ···
  }
> .gallary .card img:hover{ ···
  }

> .about{ ···
```

```
</section>

<section class="gallary" id="gallary">
  <div class="container">

    <div class="main-txt">
      <h1><span>G</span>allary</h1>
    </div>

    <div class="row" style="margin-top: 30px;">
      <div class="col-md-4 py-3 py-md-0">
        <div class="card">
          <img src="./images/g1.png" alt="" height="230px">
        </div>
      </div>
      <div class="col-md-4 py-3 py-md-0">
        <div class="card">
          <img src="./images/g2.png" alt="" height="230px">
        </div>
      </div>
      <div class="col-md-4 py-3 py-md-0">
        <div class="card">
```





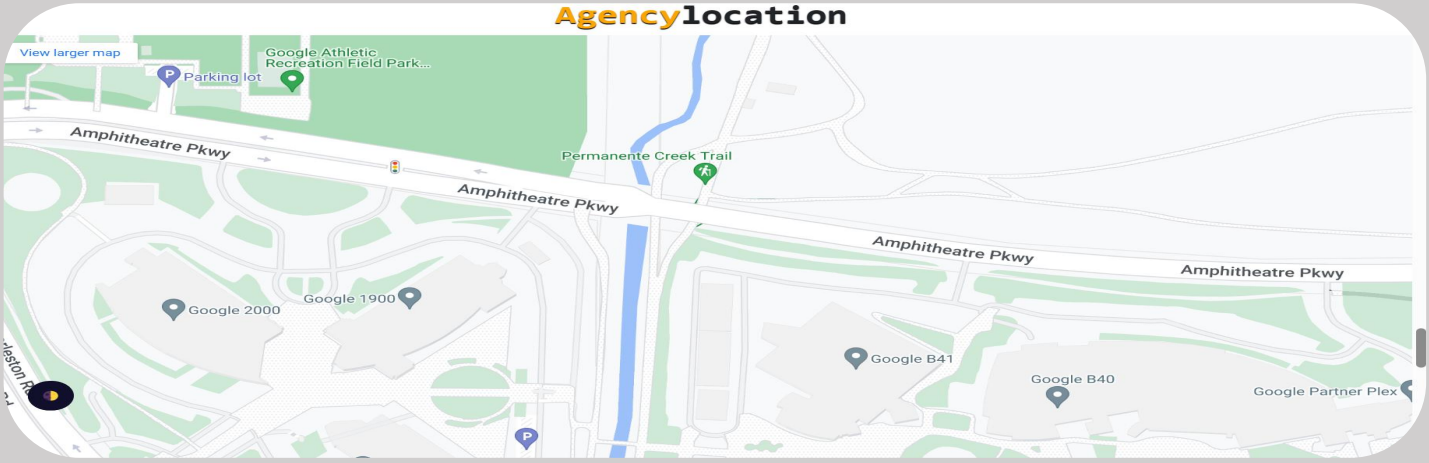
Section 5, which showcases information about our tourism company and a promotional video about our tourism services.

--> HTML-CSS:

```html
            </div>
          </div>

        </div>
      </section>
      <section class="about" id="about">
        <div class="container">

          <div class="main-txt">
            <h1>About <span>Us</span></h1>
          </div>

          <div class="row" style="margin-top: 50px;">

            <div class="col-md-6 py-3 py-md-0">···
            </div>

            <div class="col-md-6 py-3 py-md-0">···
```

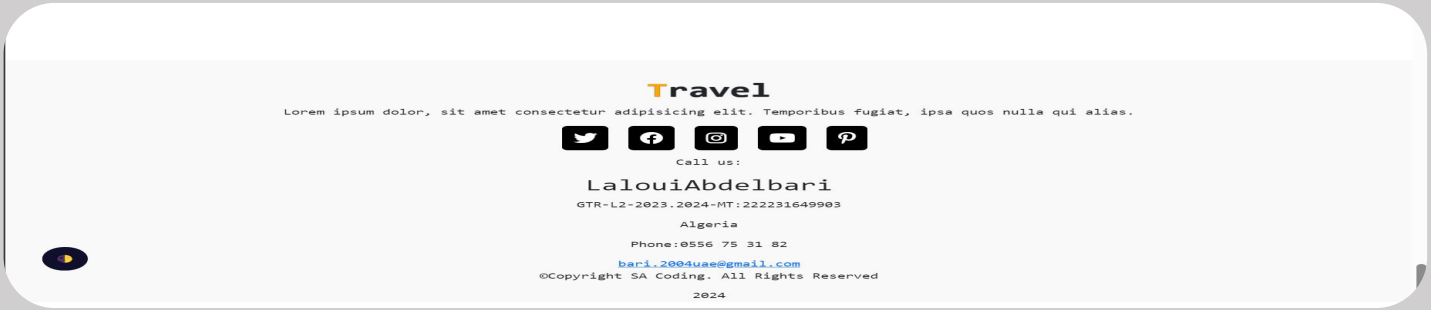
```css
.about{···
}
.about .card{···
}
.about .card img{···
}
.about h2{···
}
.about p{···
}
#about-btn{···
}
#about-btn:hover{···
}
@media (max-width:765px){···
```

Agencylocation



Section 6 displays the location of our agency, implemented using JS and a link.

[Footer]



(9)

The footer of the page displays copyright information, important details for website administrators, and our contact accounts for communication.

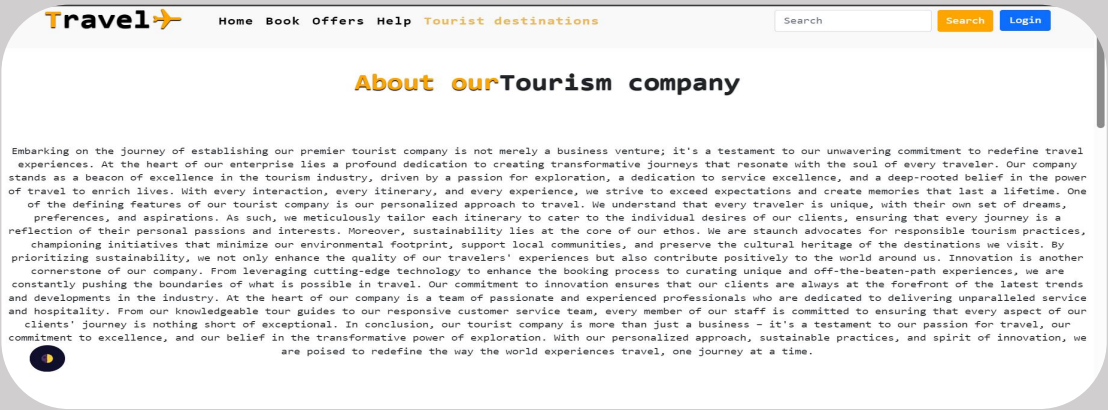
--> HTML-CSS:

```
> #footer{ ···
  }
> #footer h1{ ···
  }
> #footer h1 span{ ···
  }
> .social-links i{ ···
  }
> .social-links i:hover{ ···
```

```
> <section class="m"> ···
  </section>
    <footer id="footer">
      <h1><span>T</span>ravel</h1>
      <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Temporibus fugiat, ipsa quos nulla qui alias.</p>
>     <div class="social-links">···
      </div>
      <div class="credit">
        <p>Call us: </p><h2>Laloui<span>Abdelbari</span></h2>
        <p>GTR-L2-2023.2024-MT:222231649903</p>
        <p>Algeria</p>
            <p>Phone:0556 75 31 82</p>
            <a href="mailto:bari.2004uae@gmail.com" class="footer-link">bari.2004uae@gmail.com</a>

      </div>
      <div class="copyright">···
```

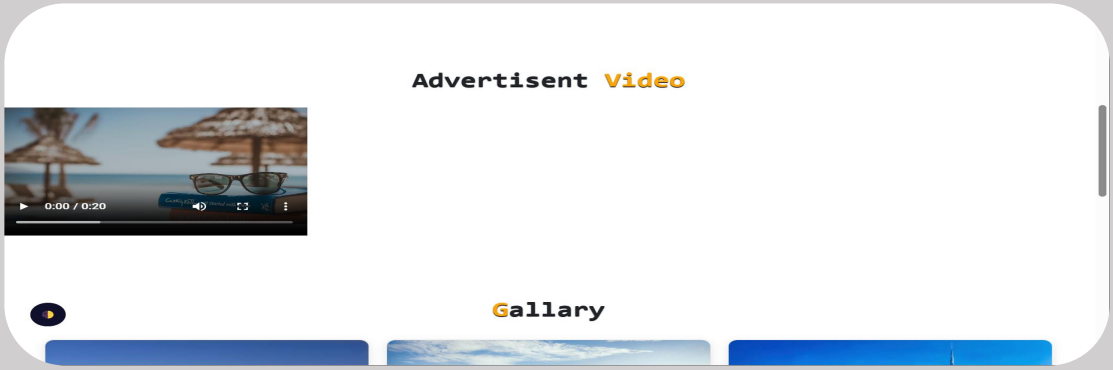
The <footer> command was used, along with 'mailto,' to create a footer section and include contact links.

About Us



(10)

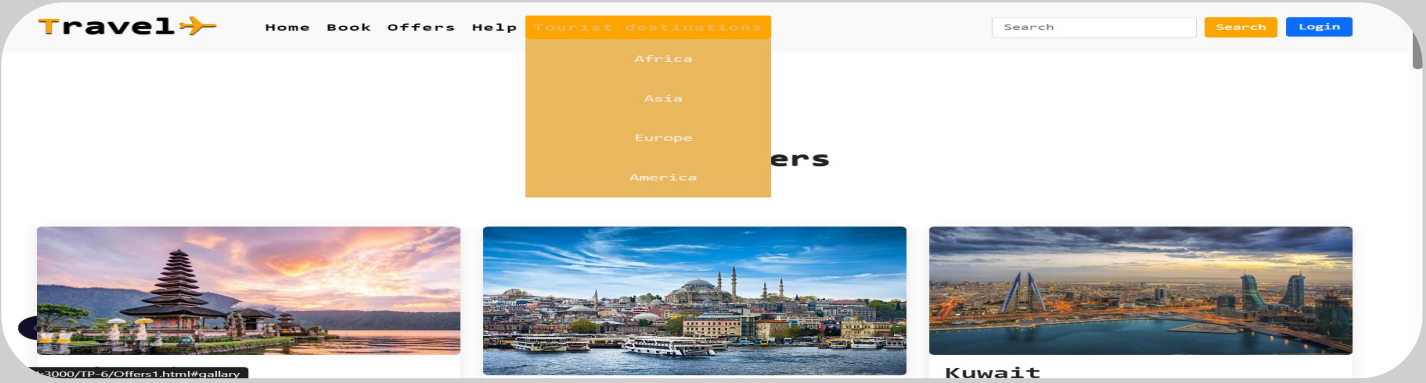
A detailed description and an elaborate explanation of the features of our leading tourism company.



(11)

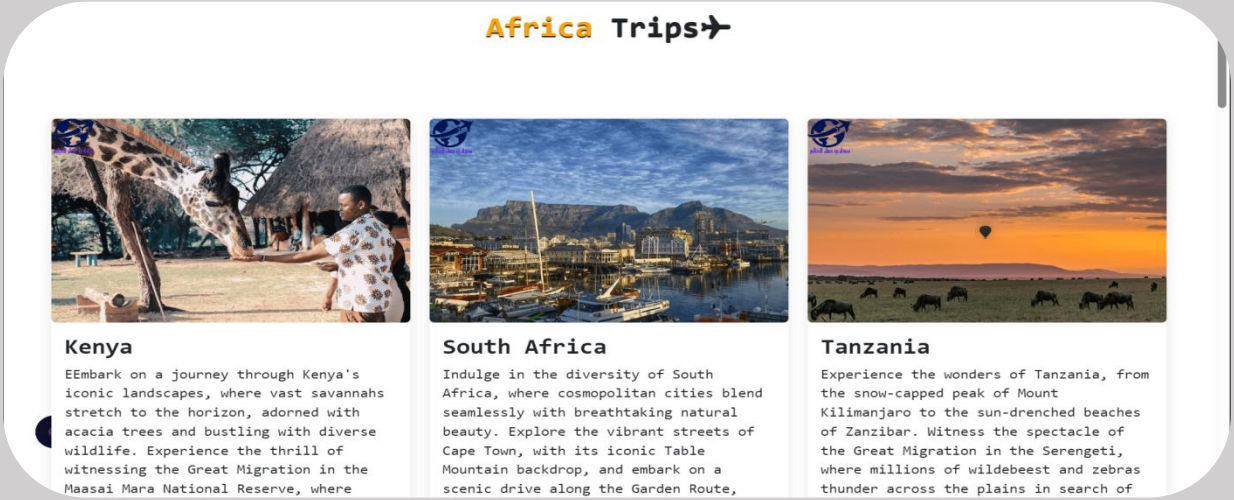
Here is a promotional video for our tourism agency (using the <video> command).

[Tourist Destinations]



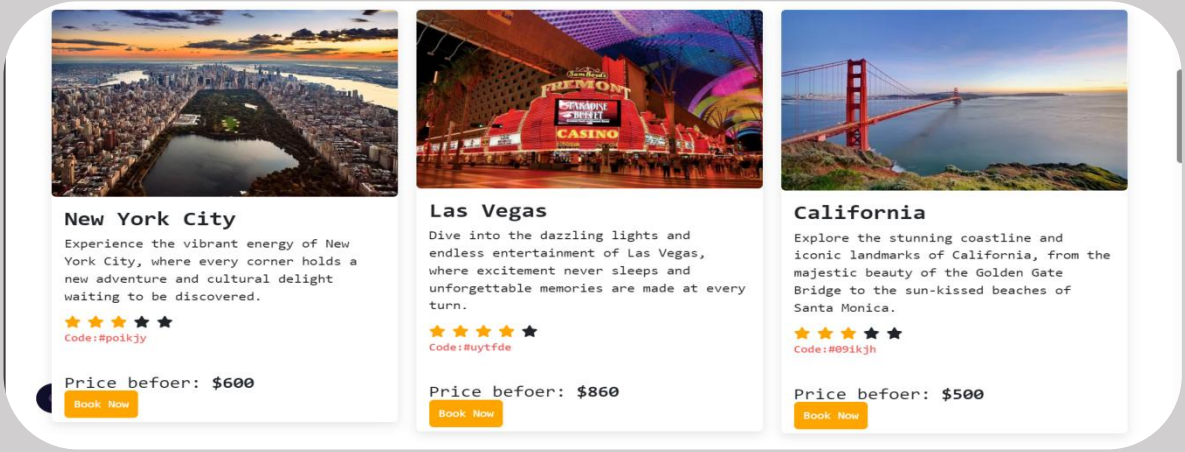
(12)

Submenus of the main menu (Tourist Destinations) display the continents where our agency offers destinations.



(13)

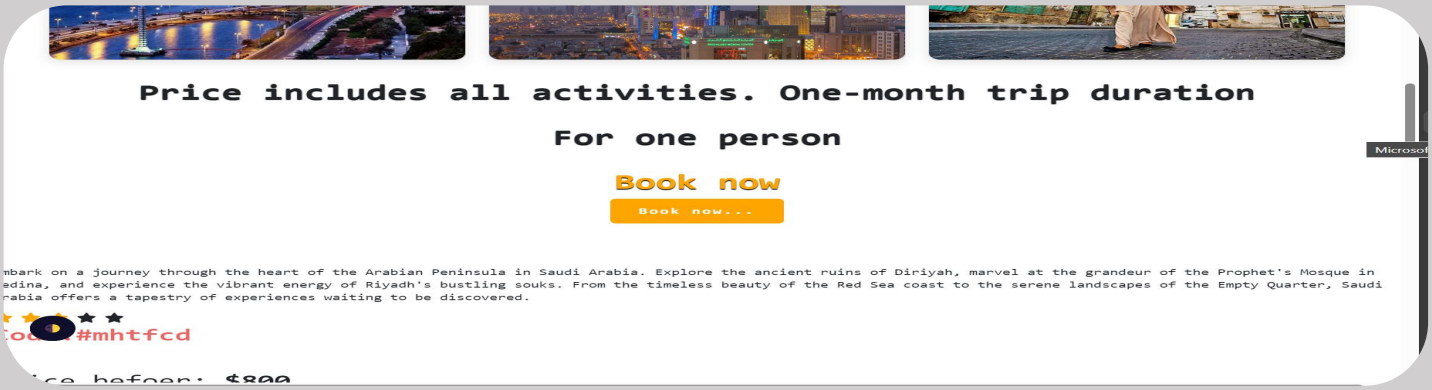
Here is an example of our exclusive packages offered in the African continent.



(14)

Here are our destinations in the continent of America.

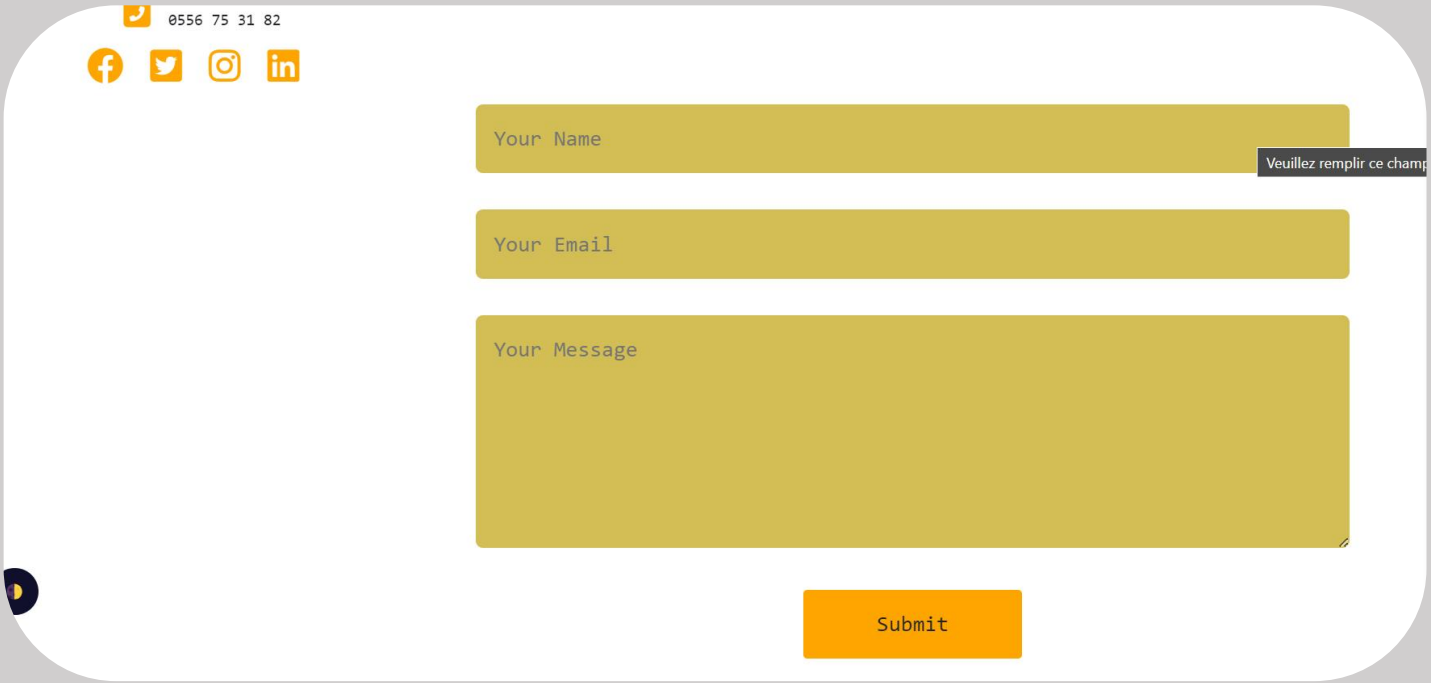
[Show Details]



(15)

When you click on the button below each of our packages, it takes you to a description of the tourist destination, where you can click to make a reservation.

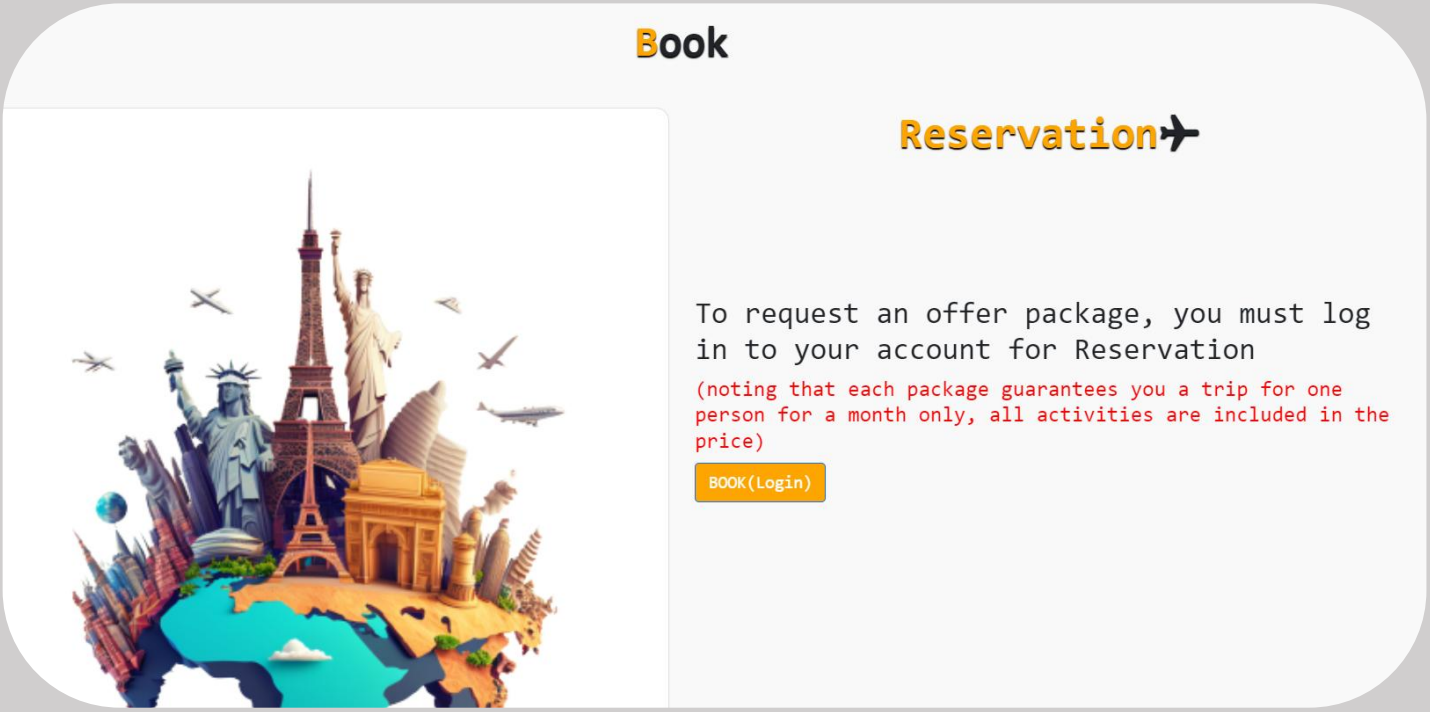
[Comment]



(15)

To enhance credibility, we provide a comment box below each package for feedback and opinions.

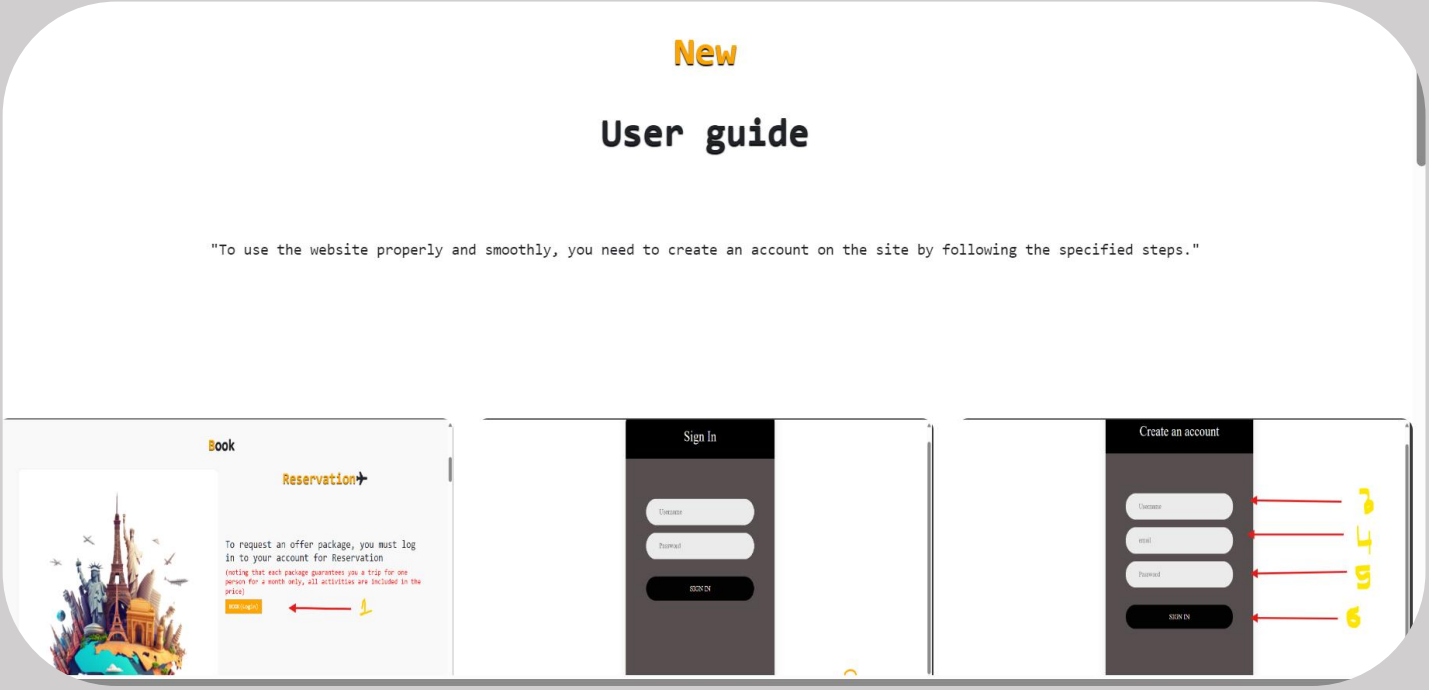
[Book]



(16)

When you click on 'book,' you are directed to a section containing important notes regarding the offer provided by the agency (please read carefully). When you click on 'request,' you will be taken to create an account for security and customer credibility.

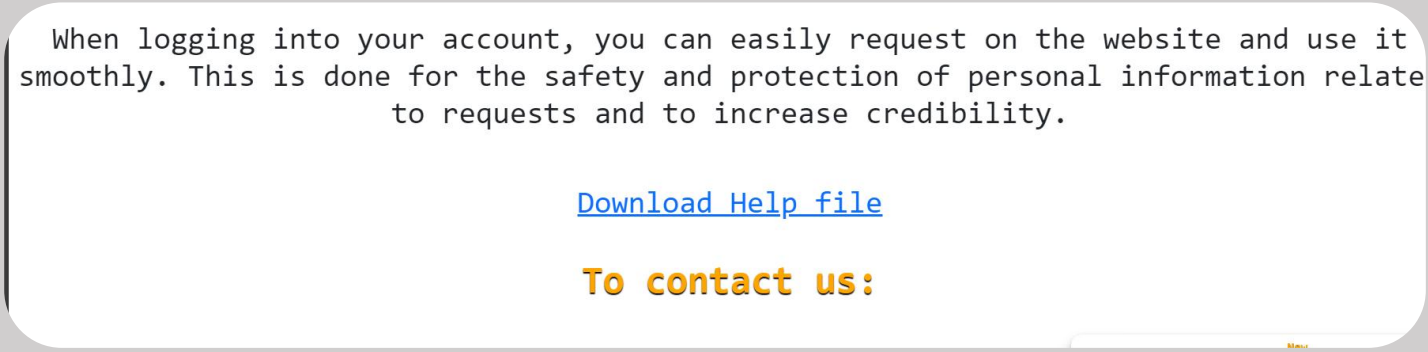
[Help]



(17)

Here is a help section where explanatory images are displayed, showing descriptions when hovered over.

- "The information provided specifically details how to create an account."

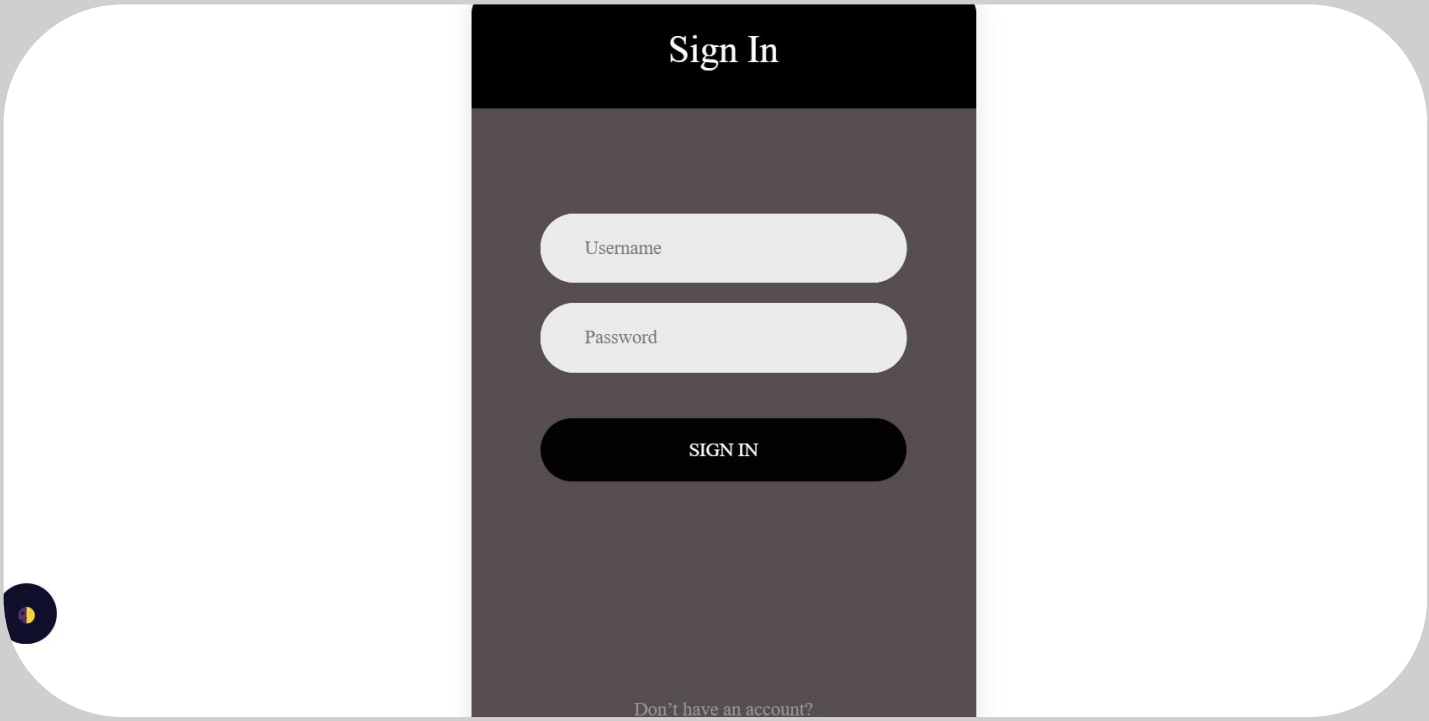


(18)

At the bottom of the images section, there is a download link for an explanatory file detailing how the website works.

{ Login To Accounts}

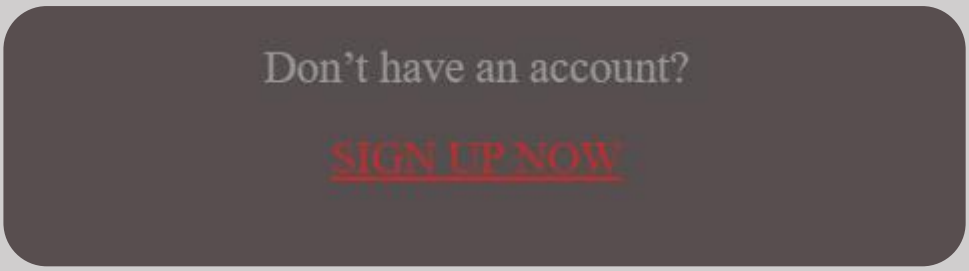
[Sign Ln]



(1)

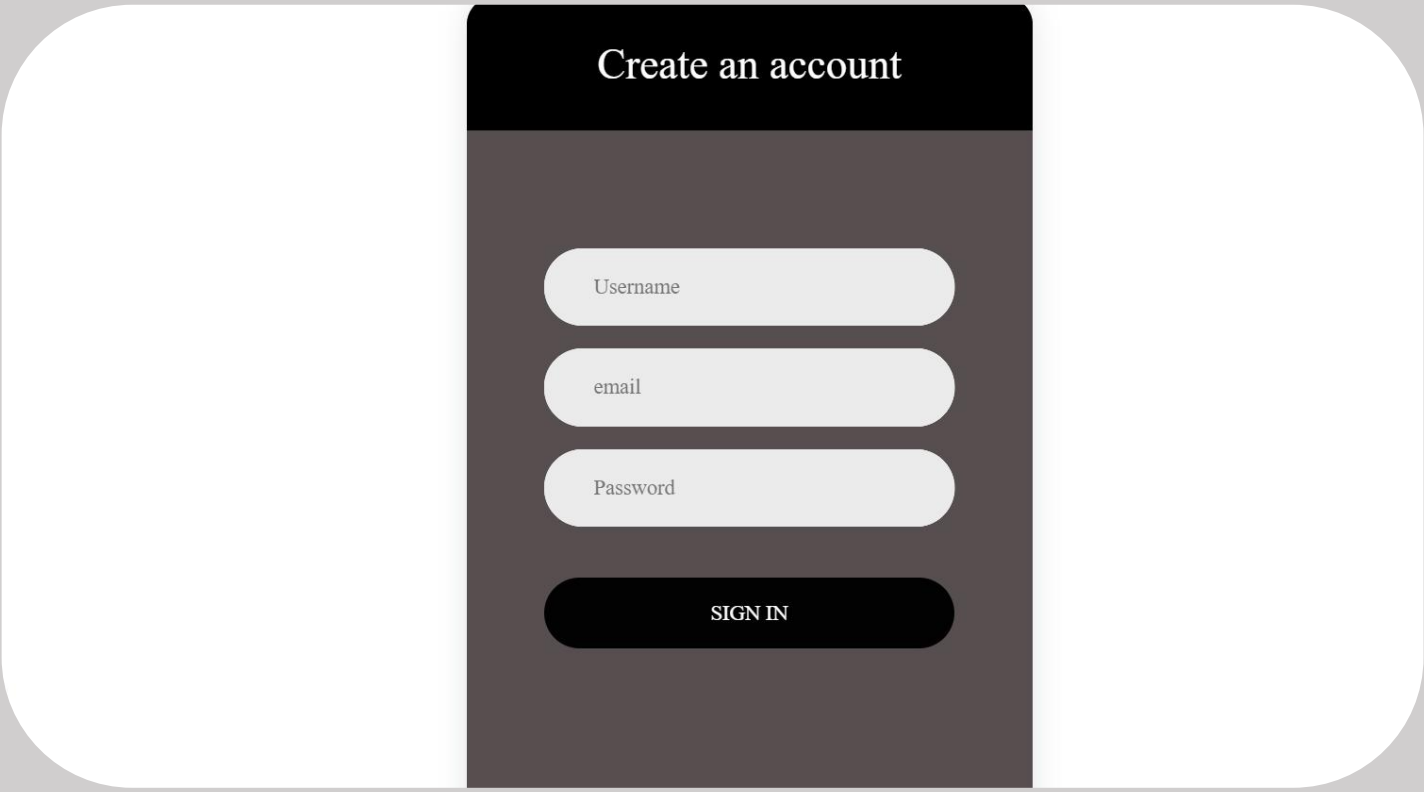
Enter your account by providing a username and password. Additionally, access the admin account through a special login (not changed via command).

[Sign up now]



(2)

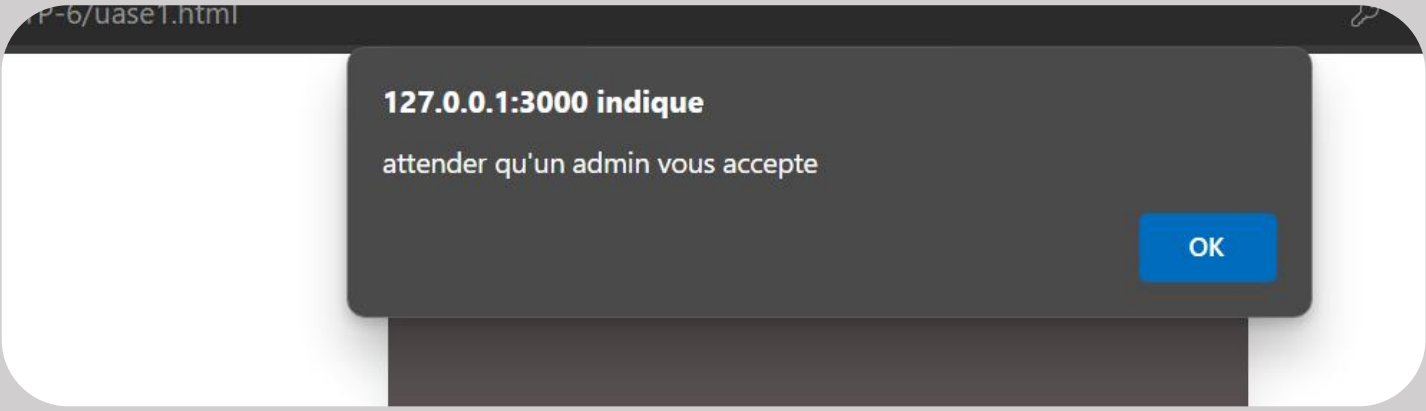
If you don't have an account, click on 'Sign Up Now'.



(3)

Enter your username, password, and email.

[Admin Approval]



(4)

When you click on 'Sign In,' a message will appear indicating that you need to wait for

admin approval before you can access your account.

--> JS:

```
ody>
<script>const options = {
const darkmode = new Darkmode(options);

darkmode.showWidget();

</script>
add a breakpoint

<script >const admin = {
    name: "admin",
    mdp: "GTR",
  };

  const user_name = document.getElementById("user_name");
  const user_mdp = document.getElementById("user_mdp");
  const btn = document.getElementById("btn");

  btn.addEventListener("click", function (e) {
    e.preventDefault();

    if (user_name.value == admin.name && user_mdp.value == admin.mdp) {
      location.href = "admin1.html";
      user_name.style.color = "green";
```

OUTPUT DEBUG CONSOLE TERMINAL PORTS Filter (e.g. text, !exclude)

```
avaScript API warning: InvalidKey https://developers.google.com/maps/documentation/javascript/error-messages#invalid-key
```

Here is an illustration of a command to save the username and password for an admin.

```
<script>
rol (Ctrl+Shift+G)
  const user_email = document.getElementById("user_email");
  const user_mdp = document.getElementById("user_mdp");

  const btn = document.getElementById("btn");

  btn.addEventListener("click", function (e) {
    let newacc = JSON.parse(localStorage.getItem("newacc")) || [];

    e.preventDefault();
    if (
      user_name.value == "" &&
      user_email.value == "" &&
      user_mdp.value == ""
    ) {
      alert("vous devez remplire tout les champs");
    } else {
      let nacc = {
        nom_complet: user_name.value,
        email: user_email.value,
        password: user_mdp.value,
        nbrtr: [],
      };
```

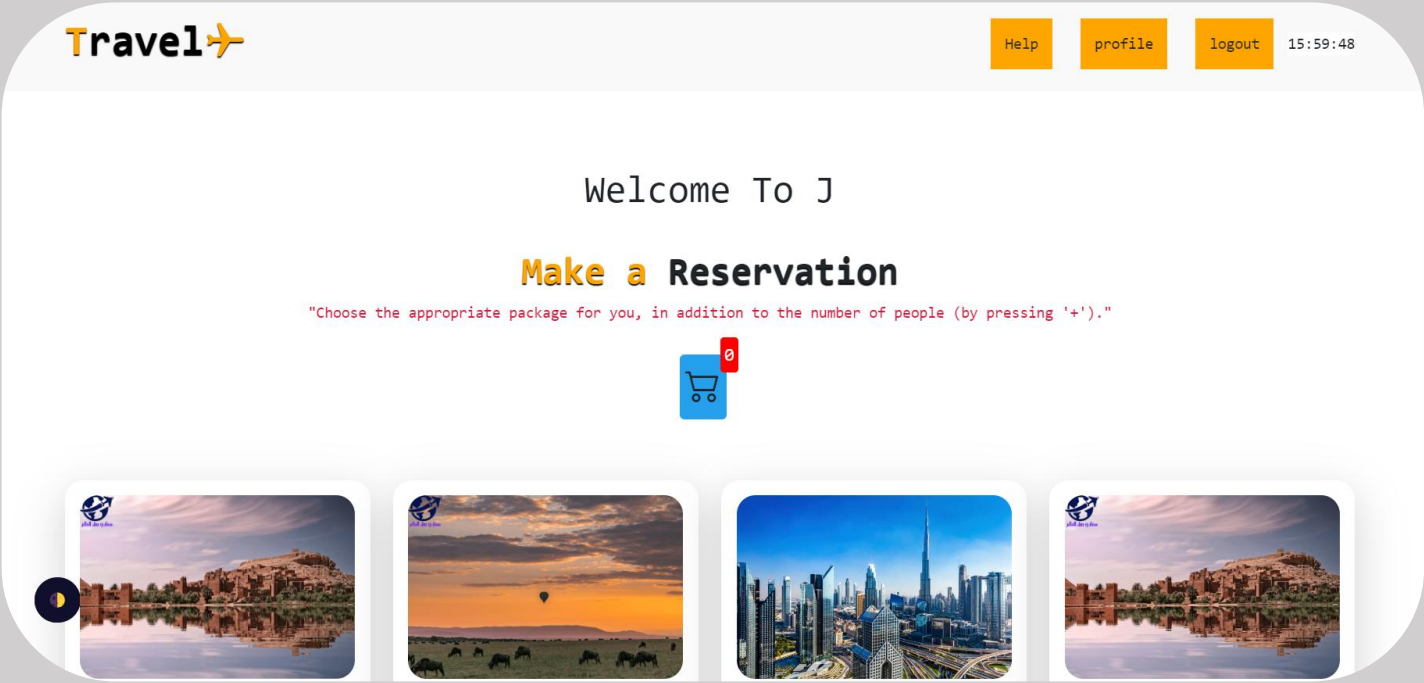
```
49        e.preventDefault();
50        if (
51          user_name.value == "" &&
52          user_email.value == "" &&
53          user_mdp.value == ""
54        ) {
55          alert("vous devez remplire tout les champs");
56        } else {
57          let nacc = {
58            nom_complet: user_name.value,
59            email: user_email.value,
60            password: user_mdp.value,
61            nbrtr: [],
62          };
63          newacc.push(nacc);
64          localStorage.setItem("newacc", JSON.stringify(newacc));
65
66          alert("attender qu'un admin vous accepte");
```

MS OUTPUT DEBUG CONSOLE TERMINAL PORTS

By using DOM (Document Object Model), many features of a user are enabled on a website, such as document.getElementById, which allows us to retrieve a specific value in the desired section by its ID.

- By using JSON and Cookies, we can save the entered data.

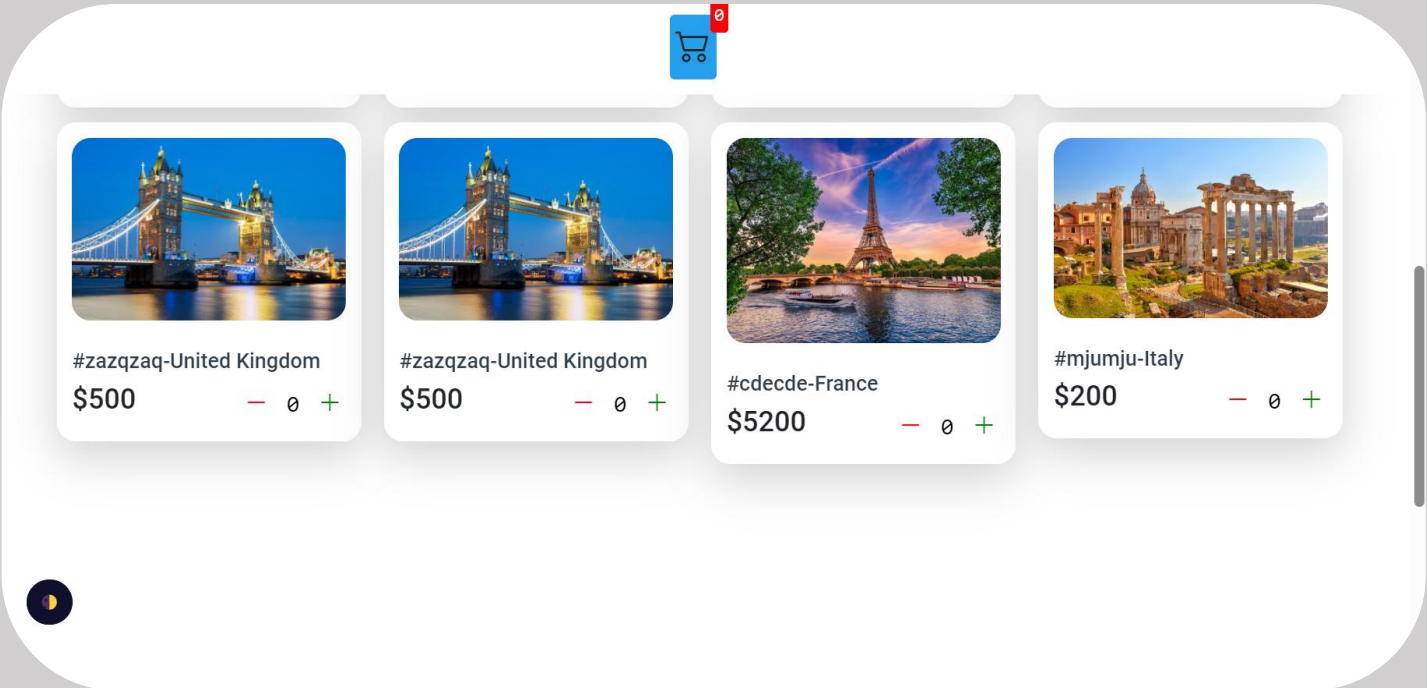
{ Uesr-Interface}



(5)

The user interface includes a help section, a profile, a log out option, and packages for request.

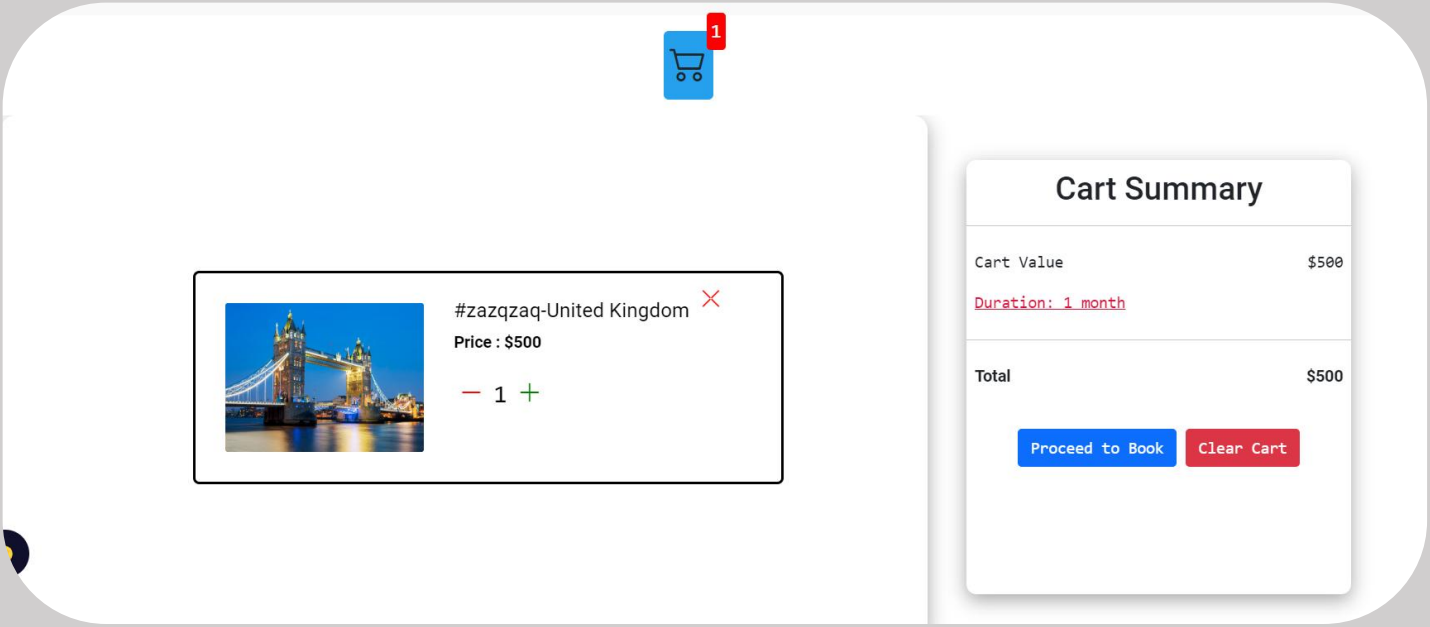
[Purchase Reservation]



(6)

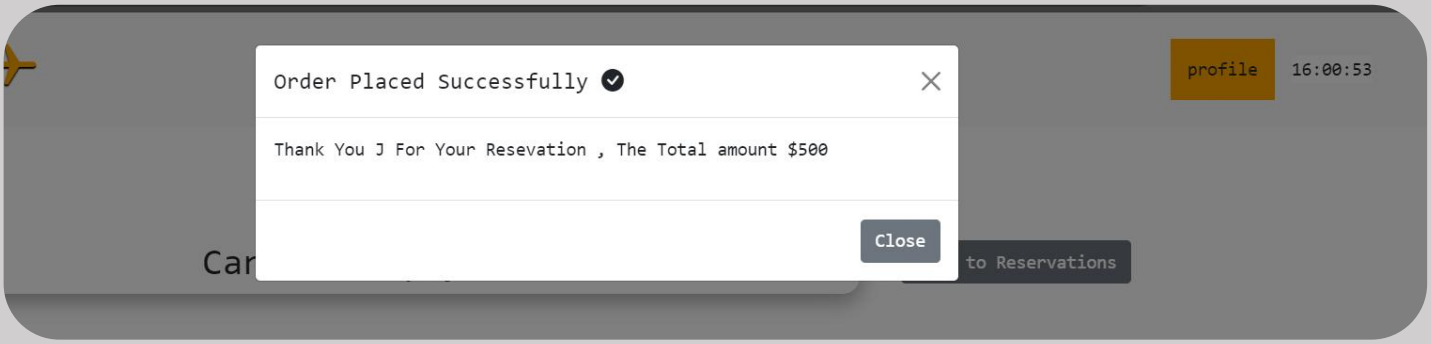
To clarify how to request/select a suitable package:

- Use the (+) button to add the number of people.



(6)

When adding a package offer, click on the basket to see a picture of the offer + price and the number of people (you can add the number of people at this stage). Confirm the request.



(7)

After confirming the request, a message will appear confirming that you have purchased a tourist offer and displaying the price.

[Personal Information]

(8)

The personal information section displays your personal details and allows you to change information that automatically updates in your database.

--> JS:

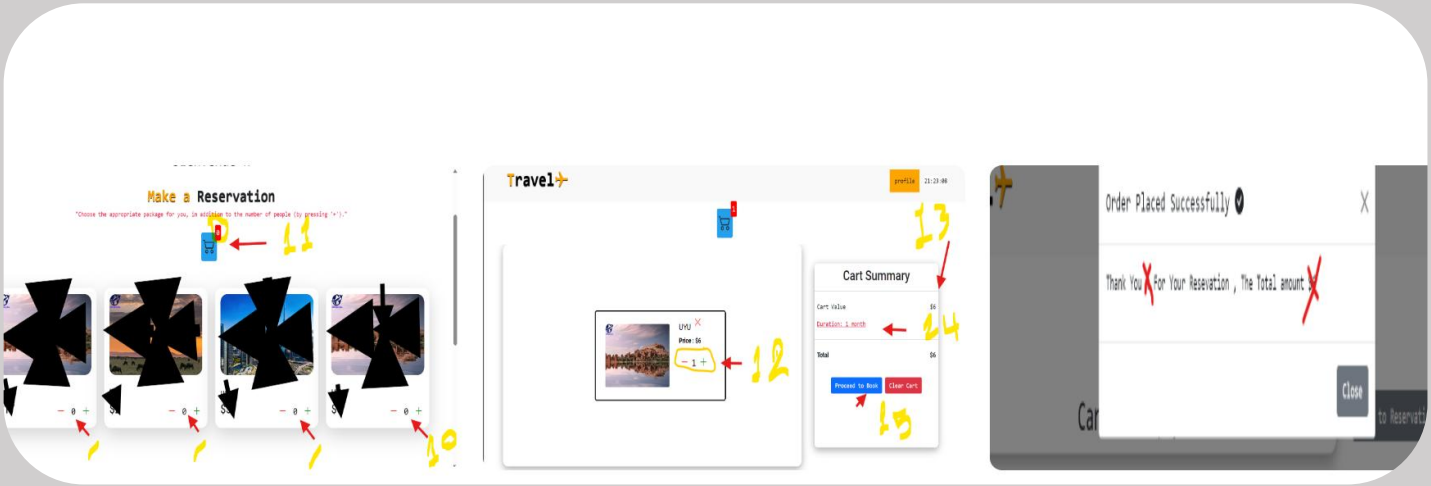
```
<script>
  let tempo = JSON.parse(localStorage.getItem("tempo"));
  let accounts = JSON.parse(localStorage.getItem("accounts")) || [];

  let nom = document.getElementById("nom");
  let email = document.getElementById("email");
  let mdp = document.getElementById("mdp");
  const mdf = document.getElementById("mdf");

  nom.innerHTML = `Nom : ${tempo.nom_complet}`;
  email.innerHTML = `Email : ${tempo.email}`;
  mdp.innerHTML = `Password : ${tempo.password}`;

  mdf.addEventListener("click", function (e) {…
</script>
  <script>…
  </script>
```

```
  <script >let shopItemsData = JSON.parse(localStorage.getItem("product")) || [];
  </script>
  <script >let shop = document.getElementById("explore-menu-section-items");
      const bienvenue = document.getElementById("bienvenue");

      let mybasket = JSON.parse(localStorage.getItem("data")) || [];

      let temp = JSON.parse(localStorage.getItem("tempo"));
      bienvenue.innerHTML = `Welcome To ${temp.nom_complet}`;
      let generateShop = () => {
        return (shop.innerHTML = shopItemsData
          .map((x) => {
            let { id, productName, img, price } = x;
            let search = mybasket.find((x) => x.id === id) || [];
            return `
                <div id=product-id-${id} class="col-12 col-md-6 col-lg-3 ">
                    <div class="shadow-lg menu-item-card p-3 mb-3 bg-white">
                    <img src=${img} class="menu-item-image w-100" />
                    <h1 class="product-card-title">${productName}</h1>
                    <div class="d-flex flex-row justify-content-between">
```

```
tml lang="en">
  </body>
  <script>let shopItemsData = JSON.parse(localStorage.getItem("product")) || [];</script>
<script>let label = document.getElementById("label");
  let ShoppingCart = document.getElementById("shopping-cart");

  let mybasket = JSON.parse(localStorage.getItem("data")) || [];
  let temp = JSON.parse(localStorage.getItem("tempo"));

  let totalCartQuantity = () => {
    let cartQuantity = document.getElementById("cartQuantity");
    cartQuantity.innerHTML = mybasket
      .map((x) => x.item)
      .reduce((x, y) => x + y, 0);
  };

  totalCartQuantity();

  let generateCartItems = () => {
    if (mybasket.length !== 0) {
      return (ShoppingCart.innerHTML = mybasket
        .map((x) => {
          let { id, item } = x;
          let search = shopItemsData.find((y) => y.id === id) || [];
          let { img, price, productName } = search;
```

Using JSON and saving data in variables for each account created on the page.

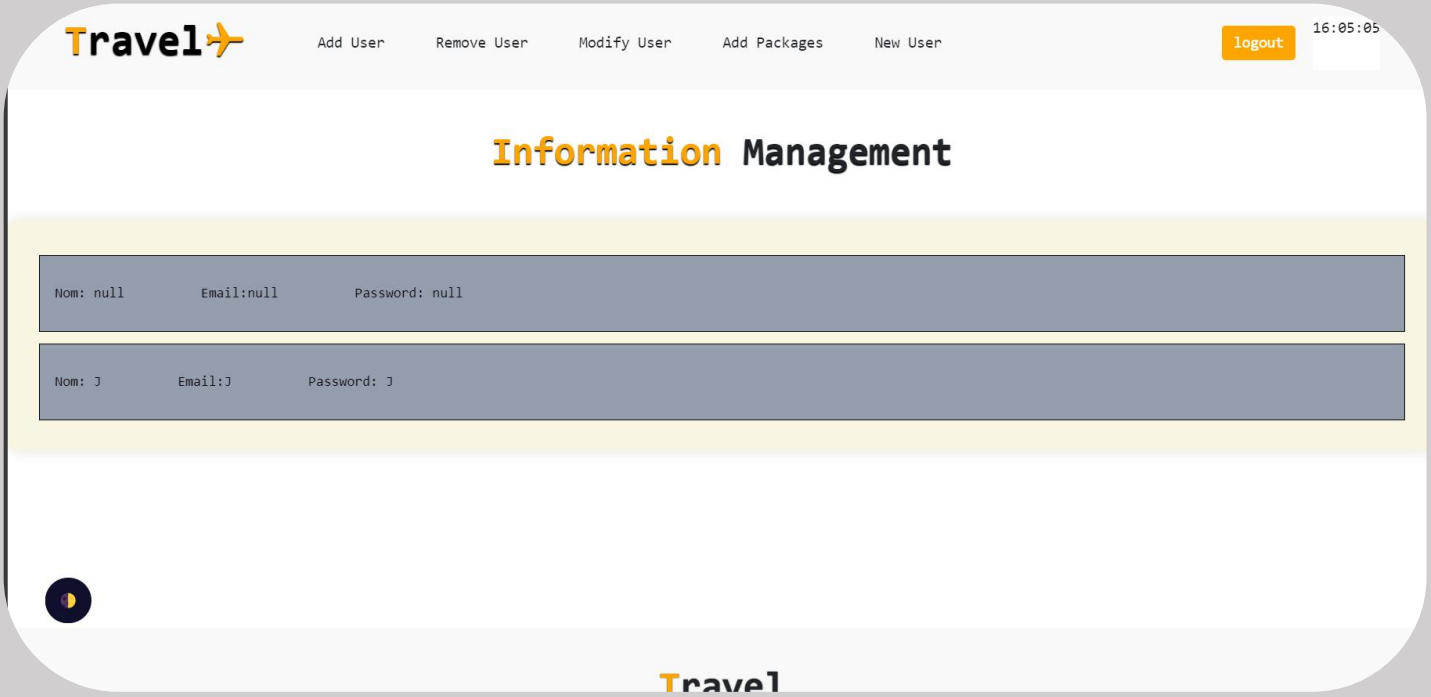
[Assistance after account registraion]



(9)

In the help section of the user account interface, the process of requesting on the website is displayed and explained.

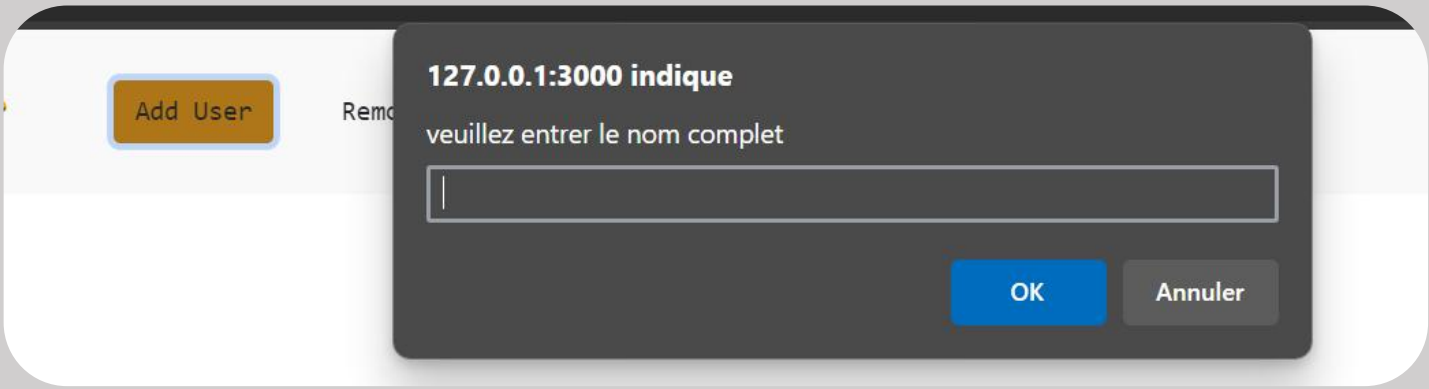
{ Admin-Interface}



(1)

In the admin interface, user information is displayed, including name, number, and user account. Options include adding, deleting, and editing user information, as well as adding offers and approving account entries on the website.

[Add User]

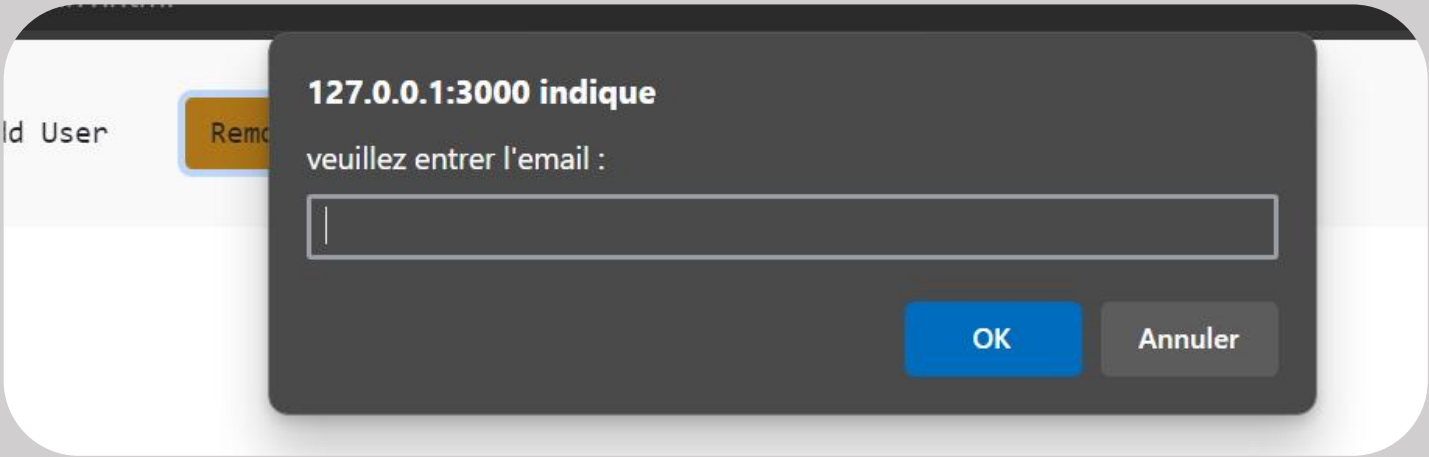


(2)

Steps to add a user by entering their first name, last name, and email:

1. Navigate to the admin interface.
2. Find the option to add a user.
3. Enter the user's first name.
4. Enter the user's last name.
5. Enter the user's email.
6. Save or submit the information to add the user.

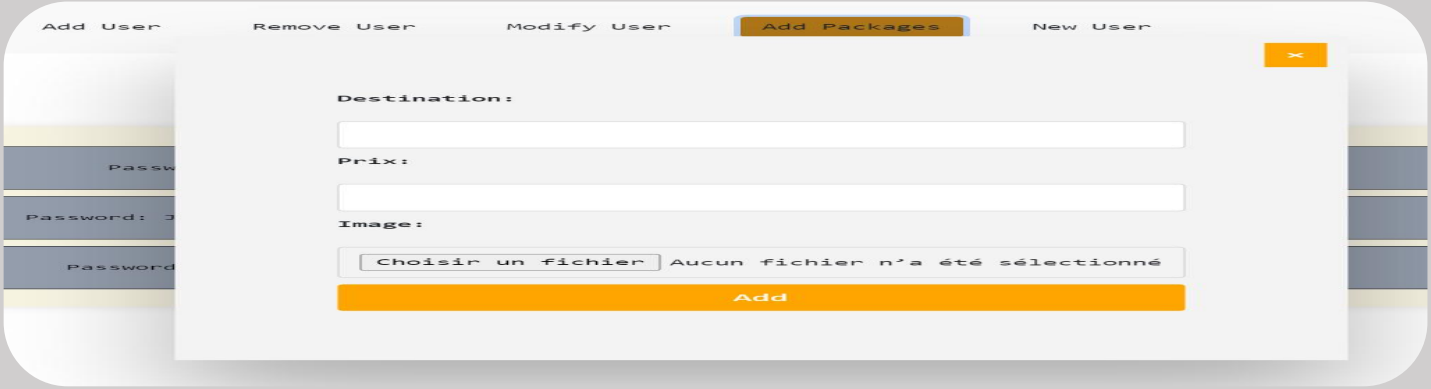
[Delete User]



(3)

To delete a user, you need to enter their name, password, and email.

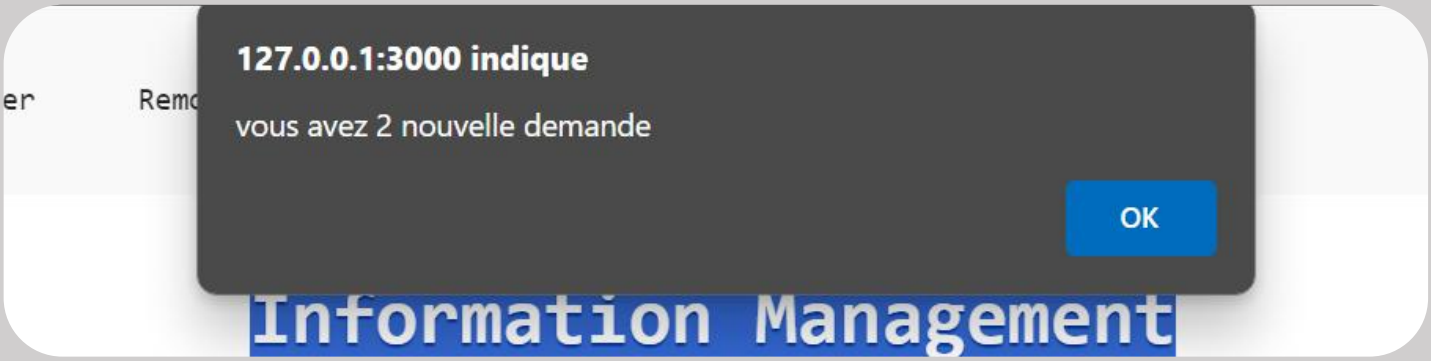
[Add Product]



(4)

To add a product, you need to enter the destination name, destination code, destination symbol, price, and image.

[Edit User]



(5)

Here, approval is required when requesting to create an account on the website.

--> JS:

```
</script>


<script >const btn = document.querySelectorAll(".btn");
    const info = document.getElementById("info");
    const modal_item = document.querySelector(".modal_item"
    const add_item = document.getElementById("add_item");
    const egle9 = document.getElementById("egle9");

    egle9.addEventListener("click", function (e) {
      e.preventDefault();
      modal_item.classList.add("hidden");
    });

    let i = 0;
    let accounts = JSON.parse(localStorage.getItem("account

    btn.forEach((btns) => {
      btns.addEventListener("click", function (e) {
        e.preventDefault();
```

```
body>
<script >const btn = document.querySelectorAll(".btn");
    btn.forEach((btns) => {
      btns.addEventListener("click", function (e) {
        if (btns.value == "add") {
          let nom_complet = prompt("veuillez entrer le nom complet");
          let email = prompt("veuillez entrer l'email");
          let mdp = prompt("veuillez entrer un mdps");
          if (nom_complet == "" && email == "" && mdp == "") {
            alert("le champ est vide");
          } else {
            const account = {
              nom_complet: nom_complet,
              email: email,
              password: mdp,
              nbrtr: [],
            };
            let accounts = JSON.parse(localStorage.getItem("accounts"))
            accounts.push(account);
            localStorage.setItem("accounts", JSON.stringify(accounts));
            diplay();
            location.reload();
```

```
    }


    if (btns.value == "nv_dmn") {
      let accounts = JSON.parse(localStorage.getItem("accounts")) || [];

      let newacc = JSON.parse(localStorage.getItem("newacc")) || [];
      alert(`vous avez ${newacc.length} nouvelle demande`);
      newacc.forEach((acc, i) => {
        alert(`demande n° ${i + 1}: ${acc.nom_complet}`);
        let com = prompt(
          " ajouter (oui) refus (non)"
        );

        if (com == "oui") {
          accounts.push(acc);
          newacc.splice(i, 1);
        }
        localStorage.setItem("newacc", JSON.stringify(newacc));
```

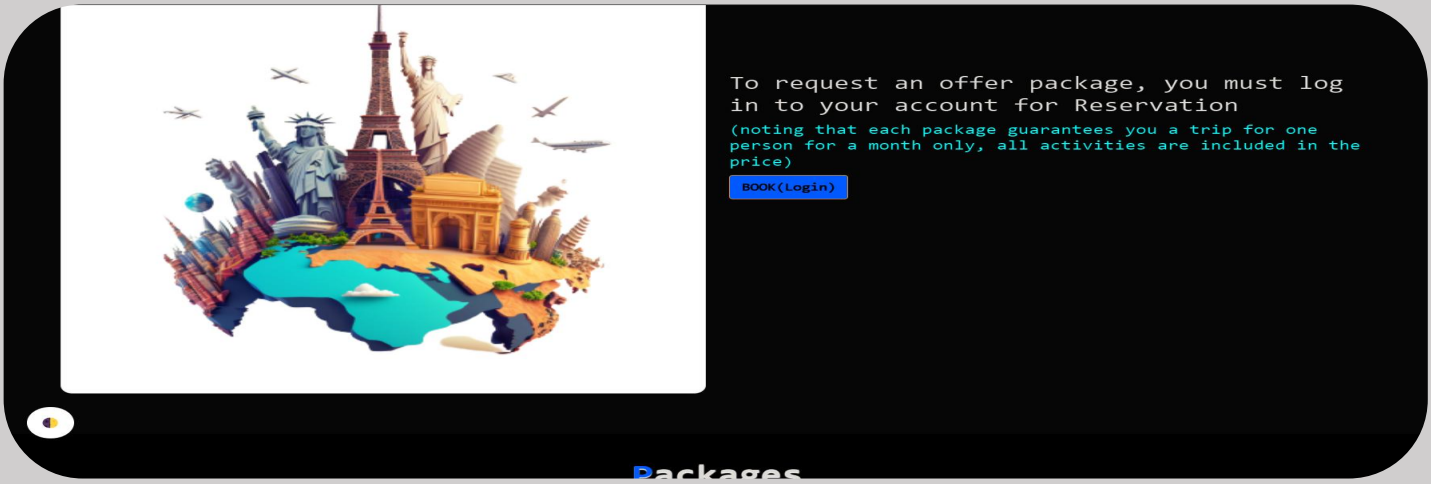
```
if (btns.value == "mod") {
  let accounts = JSON.parse(localStorage.getItem("accounts")) || [];

  let name_edit = prompt("user name");
  let mdp_edit = prompt("mdps");
  console.log(name_edit, mdp_edit);
  accounts.forEach((acc) => {
    if (acc.nom_complet == name_edit && acc.password == mdp_edit) {
      let nom = prompt("new username");
      let email = prompt("new email");
      let mdp = prompt("new mdps");
      acc.nom_complet = nom;
      acc.email = email;
      acc.password = mdp;
    }
  });
  localStorage.setItem("accounts", JSON.stringify(accounts));
```

```
<script >const btn = document.querySelectorAll(".btn");
  btn.forEach((btns) => {
    btns.addEventListener("click", function (e) {


      if (btns.value == "des") {
        location.href = "b.html";
      }
    });
  });

  function diplay() {
    let accounts = JSON.parse(localStorage.getItem("accounts")) || [];

    accounts.forEach((acc) => {
      html = `
      <div class="info_user">
            <p>Nom: ${acc.nom_complet}</p>
            <p>Email:${acc.email}</p>
            <p>Password: ${acc.password}</p>
      </div>
    `;
```

{ Additional Benefits}

[Change Theme]

(1)



(2)

The website has the feature of changing the theme for all sections of the website.

--> JS:

```
<script>const options = {
  bottom: '64px', // default: '32px'
  right: 'unset', // default: '32px'
  left: '32px',// default: 'unset'
  time: '0.5s',// default: '0.3s'
  mixColor: '#fff',// default: '#fff'
  backgroundColor: '#fff',  // default: '#fff'
  buttonColorDark: '#100f2c',  // default: '#100f2c'
  buttonColorLight: '#fff',// default: '#fff'
  saveInCookies: false , // default: true,
  label: '◑' ,// default: ''
}
const darkmode = new Darkmode(options);

darkmode.showWidget();

</script>
```

Using JavaScript to provide appropriate colors for displaying a cohesive theme.

[Clock]

(3)

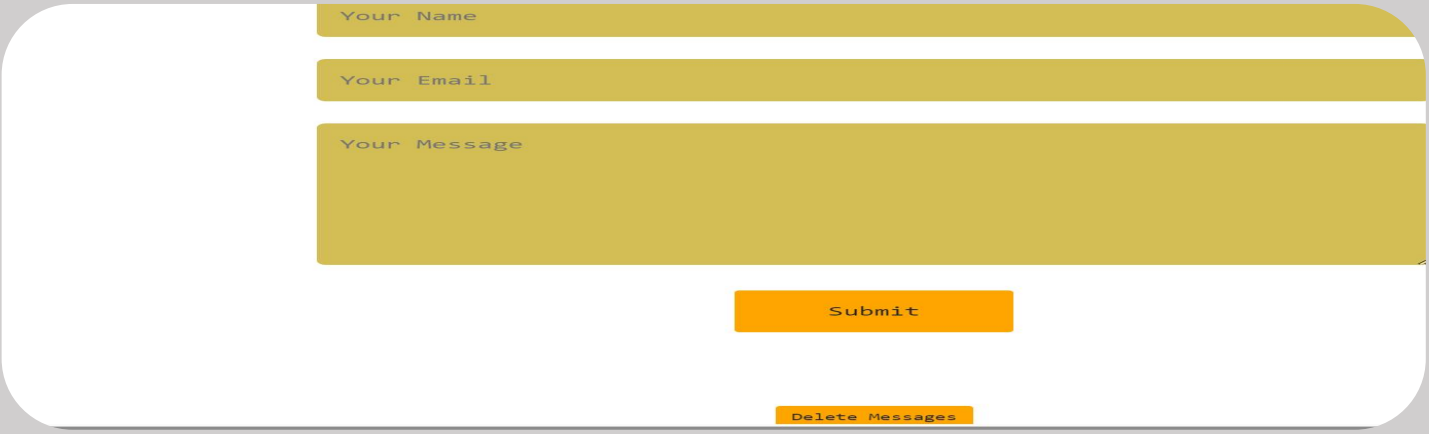
The user interface and admin interface are displayed at the top of the page.

--> JS:

```html
<html lang="en">
<body>
<script >const btn = document.querySelectorAll(".btn");
diplay();
</script>
<script>
function updateClock() {
const now = new Date();
const hours = now.getHours().toString().padStart(2, '0');
const minutes = now.getMinutes().toString().padStart(2, '0');
const seconds = now.getSeconds().toString().padStart(2, '0');
const timeString = `${hours}:${minutes}:${seconds}`;
document.getElementById('clock').textContent = timeString;
}

// Update the clock every second
setInterval(updateClock, 1000);

// Initial call to display the clock immediately
updateClock();
</script>
</body>
</html>
```

Using JavaScript, a synchronized clock is created to match the device's clock.

[Comment]



(4)

The feature of adding comments.

--> JS:

```javascript
}

// Event listener for form submission
document.getElementById('contactForm').addEventListener('submit', function(event) {
  event.preventDefault();
  var formData = new FormData(this);
  setCookie('Name', formData.get('Name'), 30); // Save name in cookie for 30 days
  setCookie('Email', formData.get('Email'), 30); // Save email in cookie for 30 days
  setCookie('Message', formData.get('Message'), 30); // Save message in cookie for 30 days
  displaySavedData(); // Update displayed saved data
  this.reset(); // Reset form after submission
});

// Function to delete messages
function deleteMessages() {
  document.cookie = 'Name=; expires=Thu, 01 Jan 1970 00:00:00 UTC; path=/;';
  document.cookie = 'Email=; expires=Thu, 01 Jan 1970 00:00:00 UTC; path=/;';
  document.cookie = 'Message=; expires=Thu, 01 Jan 1970 00:00:00 UTC; path=/;';
  document.getElementById('savedData').innerHTML = ''; // Clear displayed messages
}

// Event listener for delete messages button
```

Comments are saved using cookies.

[Dynamic]

(5)

A website's dynamism if it aligns with all device sizes.

--> CSS:

```
        background: #ffa500;
        color: white;
    }
    @media (max-width:850px){
        .home{
            background-position: 50%;
        }

    }
    @media (max-width:450px){ ···
    }
    /* Home Section End */

    /* Section Book Start */
```

Using CSS for media queries to define properties and change them when certain size conditions are met.

{website maintenance}

The measures taken for maintenance by an administrator, who has the ability to manage accounts, enabling them to control users' accounts by altering or deleting information. Additionally, they can submit and place new offers on a website to display them on users' accounts.
Maintenance measures include monitoring user accounts for any unauthorized changes, regularly updating security protocols to prevent hacking attempts, conducting audits to detect any irregularities or suspicious activities, and providing regular training to administrators on security best practices. Additionally, implementing multi-factor authentication, encryption, and access controls can enhance the overall security posture of the system. Regular backups of data are also essential to ensure data integrity and availability in case of any system failures or data breaches.

{ The accomplished goals }

Many objectives have been achieved, as different interfaces for the website have been created, allowing administrators, users, and visitors to interact with ease. User account data is stored securely, and products can be ordered effortlessly. The site boasts aesthetic appeal and functionality, with added packages for tourism offers, ensuring a comfortable user experience managed by the administrator.

{Aspirations}

Creating a website with a large database to add numerous capabilities and allow wide access by the administrator, along with adding many features for the user, facilitating easier interaction. This will make you a creator of a modern website with fantastic capabilities, making it competitive in the market.

{ Completed a MINI-Project}