Front-End UI/UX Mini Project

Project Submission Template

1. Title Page

- **Project Title**: Holiday Planner

Submitted By:

- *Team Members- S Sandhya*

  Riya Kiran

  Srihari
- *Roll Number- 2463047*

  2463046

  2463060
- *College- s.sandhya@btech.christuniversity.in*

  riya.kiran@btech.christuniversity.in

  sri.hari@btech.christuniversity.in

- **Course**: *Front-End UI/UX*
- **Instructor Name**: *Dhiraj*
- **Institution**: *Christ University*
- **Date of Submission**: *26/09/2025*

2. Abstract

The project aims to design and develop a simple and user-friendly Holiday Planner web application that helps users organize and manage their travel plans efficiently. The app features creating holiday itineraries, adding destinations, scheduling activities, managing budgets, and storing essential travel details in one place. It also allows editing and deleting plans, with persistent storage to retain user data for future reference. The design emphasizes a clean and aesthetic interface with calm, vacation-inspired colors, modern typography, and intuitive navigation. The layout is responsive, ensuring a seamless experience across devices, making it a practical tool for planning stress-free and enjoyable holidays.

3. Objectives

- Build a user-friendly interface applying modern and minimalist UI principles.
- Implement persistent task management using localStorage to save all tasks and their completion status.
- Provide task filters (all, active, completed) for easy task navigation.
- Use subtle visual cues (like strikethrough for completed tasks) and minimalist emojis for clarity without clutter.
- Ensure responsiveness using Bootstrap for smooth display across desktops, tablets, and mobile devices.
- Maintain an elegant neutral color palette centered around soft light brown and light blue tones.

4. Scope of the Project

- Front-end focused, with no backend required.
- Built with HTML5, CSS3, JavaScript, Bootstrap, and jQuery.
- Task data saved within localStorage for client-side persistence.
- Full responsiveness for multiple screen sizes using mobile-first design techniques.

5. Tools & Technologies Used

Tool/Technology	Purpose
HTML5	Markup and content structure
CSS3	Styling and layout management
JavaScript and jQuery	Code editor
Chrome DevTools	Testing and debugging
Bootstrap	UI components.

6. HTML Structure Overview

- The HTML structure uses semantic tags for clarity and accessibility.
- Main container holds the app with a header (app title).
- An input field and add button for new tasks.
- A list element (<ul>) that dynamically displays tasks as list items (<li>).
- Each task item includes a checkbox emoji, task text, and icons/buttons for editing and deleting.
- Filter buttons below the list for showing all/active/completed tasks.
- Footer or small note area for app info or credits if needed.

7. CSS Styling Strategy

- Use CSS variables to define neutral color palette (light brown and light blue).
- Apply consistent spacing and padding for readability and balanced layout.
- Use modern, clean sans-serif font for all text.
- Completed tasks get a subtle strikethrough and reduced opacity.
- Button and interactive elements styled minimally with subtle hover effects.
- Responsive design with Bootstrap grid and media queries for mobile scalability.
- Minimalist emoji icons sized and aligned with text for a neat look.
- Smooth CSS transitions for state changes (hover, focus, task complete).

8. Key Features

Feature	Description
Responsive Design	Adapts seamlessly to all screen sizes
Smooth Navigation	Fixed top nav with anchor links
Project Cards	Flex-based layout with hover effects
Contact Form (non-functional)	Placeholder layout for inputs and button
Accessible Fonts & Colors	High contrast and readable typography

9. Challenges Faced & Solutions

Challenge	Solution
Ensuring task data persists smoothly without backend support.	Utilized browser localStorage API to save and load tasks
Responsive design across devices	Leveraged Bootstrap’s grid and CSS media
Managing dynamic updates to the DOM	Used jQuery

10. Outcome

- Successfully built an aesthetically pleasing, minimalistic to-do list app with all required task management features.
- Tasks are saved persistently and UI updates reflect task status changes intuitively.
- App performs well and is user-friendly on both desktop and mobile devices.
- The design is calming and professional, meeting the visual expectations of neutral color schemes.

Future Enhancements:

- Add user authentication for multi-device syncing using online storage or backend.
- Implement drag-and-drop task reordering.
- Add due dates and reminders with notifications.
- Include dark mode toggle for user preference.
- Add task categories or tags for better organization.

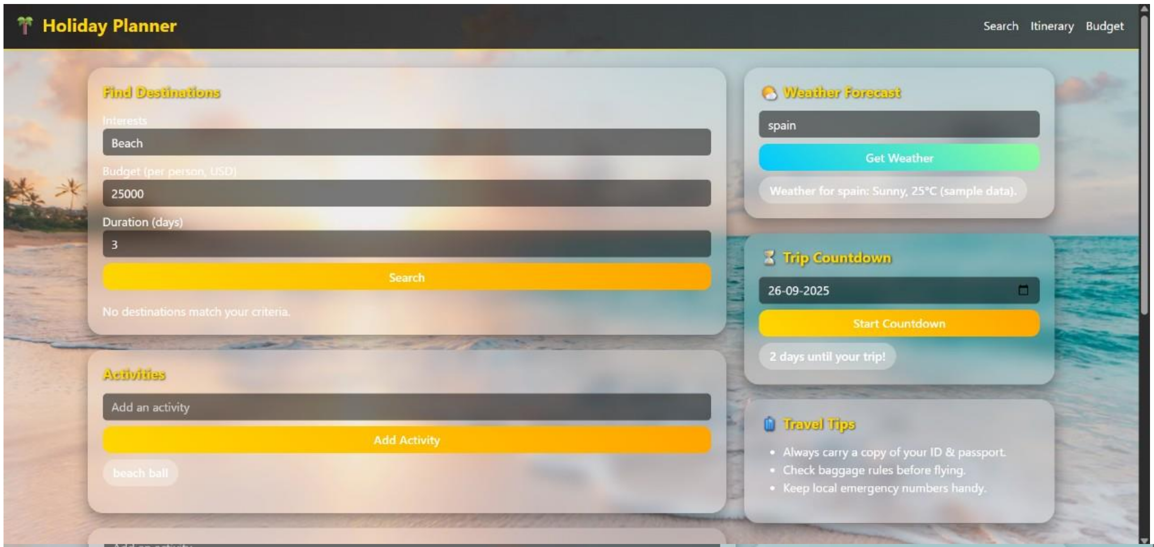
12. Sample Code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Holiday Planner</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">

  <style>
    body {
      background: url('https://images.unsplash.com/photo-1507525428034-
b723cf961d3e?auto=format&fit=crop&w=1600&q=80') no-repeat center center fixed;
      background-size: cover;
      min-height: 100vh;
      font-family: 'Segoe UI', sans-serif;
      color: #fff;
    }

    /* Glassmorphism style */
    .glass-card {
      background: rgba(255, 255, 255, 0.15);
      border-radius: 20px;
      backdrop-filter: blur(10px);
      padding: 20px;
      margin-bottom: 20px;
      box-shadow: 0 8px 20px rgba(0, 0, 0, 0.4);
    }

    .holiday-header {
      font-size: 1.2rem;
      font-weight: 700;
      margin-bottom: 15px;
      color: #FFD700;
      text-shadow: 1px 1px 3px rgba(0,0,0,0.8);
    }

    .navbar {
      background: linear-gradient(90deg, rgba(0,0,0,0.8), rgba(30,30,30,0.7));
      border-bottom: 2px solid #FFD700;
    }

    .navbar-brand {
      font-size: 1.5rem;
      font-weight: bold;
      color: #FFD700 !important;
    }

    /* Form inputs */
    .form-control, .form-select {
      background: rgba(0,0,0,0.5);
      border: none;
      color: #fff;
    }
    .form-control::placeholder {
      color: rgba(255,255,255,0.7);
    }
    .form-select option {
      background: #111;
      color: #fff;
    }

    /* Gradient buttons */
    .btn-gradient {
      border: none;
      font-weight: 600;
      border-radius: 12px;
      color: #fff;
      transition: transform 0.2s ease;
    }
    .btn-gradient:hover {
      transform: translateY(-2px);
    }
    .btn-yellow { background: linear-gradient(45deg, #FFD700, #FFA500); }
    .btn-teal { background: linear-gradient(45deg, #00C9FF, #92FE9D); }
    .btn-coral { background: linear-gradient(45deg, #FF6A88, #FF99AC); }

    /* Bubble style */
    .bubble-list {
      list-style: none;
      padding-left: 0;
      display: flex;
      flex-wrap: wrap;
      gap: 8px;
    }
    .bubble-text {
      display: inline-block;
      padding: 6px 14px;
      border-radius: 25px;
      background: rgba(255,255,255,0.2);
      backdrop-filter: blur(4px);
      font-weight: 500;
      color: #fff;
    }
  </style>
</head>
<body>

  <!-- NAVBAR -->
  <nav class="navbar navbar-expand-lg navbar-dark">
    <div class="container-fluid">
      <a class="navbar-brand" href="#">🌴 Holiday Planner</a>
      <div class="collapse navbar-collapse">
        <ul class="navbar-nav ms-auto">
          <li class="nav-item"><a class="nav-link text-white" href="#">Search</a></li>
          <li class="nav-item"><a class="nav-link text-white" href="#">Itinerary</a></li>
          <li class="nav-item"><a class="nav-link text-white" href="#">Budget</a></li>
        </ul>
      </div>
    </div>
  </nav>

  <!-- MAIN CONTENT -->
  <div class="container mt-4">
    <div class="row">

      <!-- LEFT SIDE -->
      <div class="col-md-8">

        <!-- Destinations -->
        <div class="glass-card">
          <h5 class="holiday-header">Find Destinations</h5>
          <label>Interests</label>
          <select id="interestSelect" class="form-select mb-2">
            <option value="all">All</option>
            <option value="beach">Beach</option>
            <option value="mountains">Mountains</option>
            <option value="city">City</option>
          </select>
          <label>Budget (per person, USD)</label>
          <input type="number" id="budgetInput" class="form-control mb-2" placeholder="e.g. 1500">
          <label>Duration (days)</label>
          <input type="number" id="durationInput" class="form-control mb-2" placeholder="e.g. 5">
          <button class="btn btn-gradient btn-yellow w-100" onclick="searchDestinations()">Search</button>
          <div id="destinationResults" class="mt-3">No destinations selected yet.</div>
        </div>

        <!-- Activities -->
        <div class="glass-card">
          <h5 class="holiday-header">Activities</h5>
          <input type="text" id="activityInput" class="form-control mb-2" placeholder="Add an activity">
          <button class="btn btn-gradient btn-yellow w-100 mb-2" onclick="addActivity()">Add Activity</button>
          <ul id="activityList" class="bubble-list"></ul>
        </div>
        <!-- Itinerary -->
        <div class="glass-card">
          <h5 class="holiday-header">Trip Itinerary</h5>
          <input type="text" id="dayInput" class="form-control mb-2" placeholder="Day 1
    activity/plan">
          <button class="btn btn-gradient btn-teal w-100 mb-2" onclick="addItinerary()">Add to
    Itinerary</button>
          <ul id="itineraryList" class="bubble-list"></ul>
        </div>

        <!-- Budget -->
        <div class="glass-card">
          <h5 class="holiday-header">Budget Tracker</h5>
          <input type="text" id="budgetItemInput" class="form-control mb-2" placeholder="Item (e.g.,
    Flight)">
          <input type="number" id="budgetAmountInput" class="form-control mb-2" placeholder="Cost
    in USD">
          <button class="btn btn-gradient btn-coral w-100 mb-2" onclick="addBudget()">Add
    Item</button>
          <ul id="budgetList" class="bubble-list"></ul>
          <p id="totalBudget">Total: $0</p>
        </div>
      </div>

      <!-- RIGHT SIDE -->
      <div class="col-md-4">

        <!-- Weather -->
        <div class="glass-card">
          <h5 class="holiday-header"> ⛅ Weather Forecast</h5>
          <input type="text" id="cityInput" class="form-control mb-2" placeholder="Enter city (e.g.,
    Paris)">
          <button class="btn btn-gradient btn-teal w-100" onclick="getWeather()">Get Weather</button>
          <div id="weatherResult" class="mt-2">No data yet.</div>
        </div>

        <!-- Countdown -->
        <div class="glass-card">
          <h5 class="holiday-header">⏳ Trip Countdown</h5>
          <input type="date" id="tripDate" class="form-control mb-2">
          <button class="btn btn-gradient btn-yellow w-100" onclick="startCountdown()">Start
    Countdown</button>
          <div id="countdownResult" class="mt-2">No trip set.</div>
        </div>

        <!-- Tips -->
        <div class="glass-card">
          <h5 class="holiday-header">□ Travel Tips</h5>
          <ul>
```

```
            <li>Always carry a copy of your ID & passport.</li>
            <li>Check baggage rules before flying.</li>
            <li>Keep local emergency numbers handy.</li>
          </ul>
        </div>

      </div>
    </div>
  </div>

  <script>
    const destinations = [
      { name: "Maldives", type: "beach", budget: 2000, duration: 7 },
      { name: "Paris", type: "city", budget: 1500, duration: 5 },
      { name: "Swiss Alps", type: "mountains", budget: 1800, duration: 6 },
      { name: "Bali", type: "beach", budget: 1200, duration: 5 },
      { name: "Tokyo", type: "city", budget: 1700, duration: 7 },
    ];

    function searchDestinations() {
      const interest = document.getElementById("interestSelect").value;
      const budget = parseInt(document.getElementById("budgetInput").value) || Infinity;
      const duration = parseInt(document.getElementById("durationInput").value) || Infinity;

      const results = destinations.filter(d => {
        return (interest === "all" || d.type === interest) &&
               d.budget <= budget &&
               d.duration <= duration;
      });

      const resultsDiv = document.getElementById("destinationResults");
      if (results.length > 0) {
        resultsDiv.innerHTML = results.map(d =>
          `<span class="bubble-text">${d.name} - ${d.type}, $${d.budget}, ${d.duration} days</span>`
        ).join(" ");
      } else {
        resultsDiv.innerText = "No destinations match your criteria.";
      }
    }

    function addActivity() {
      const input = document.getElementById("activityInput");
      if (input.value.trim()) {
        document.getElementById("activityList").innerHTML += `<li><span class="bubble-
text">${input.value}</span></li>`;
        input.value = "";
      }
    }

    function addItinerary() {
      const input = document.getElementById("dayInput");
      if (input.value.trim()) {
        document.getElementById("itineraryList").innerHTML += `<li><span class="bubble-text">${input.value}</span></li>`;
        input.value = "";
      }
    }

    let total = 0;
    function addBudget() {
      const item = document.getElementById("budgetItemInput").value.trim();
      const amount = parseInt(document.getElementById("budgetAmountInput").value);
      if (item && amount) {
        document.getElementById("budgetList").innerHTML += `<li><span class="bubble-text">${item}: $${amount}</span></li>`;
        total += amount;
        document.getElementById("totalBudget").innerText = `Total: $${total}`;
        document.getElementById("budgetItemInput").value = "";
        document.getElementById("budgetAmountInput").value = "";
      }
    }

    function getWeather() {
      const city = document.getElementById("cityInput").value;
      if(city) document.getElementById("weatherResult").innerHTML = `<span class="bubble-text">Weather for ${city}: Sunny, 25°C (sample data).</span>`;
    }

    function startCountdown() {
      const date = document.getElementById('tripDate').value;
      if (!date) return;

      const tripDate = new Date(date);
      const now = new Date();
      const diff = tripDate.getTime() - now.getTime();
      const days = Math.ceil(diff / (1000 * 60 * 60 * 24));

      document.getElementById('countdownResult').innerHTML =
        `<span class="bubble-text">${days >= 0 ? `${days} days until your trip!` : 'Trip date has passed.'}</span>`;
    }
  </script>

</body>
</html>
```

13. Screenshots of Final Output

DESKT0P VIEW



Find Destinations

Interests: Beach

Budget (per person, USD): 25000

Duration (days): 3

No destinations match your criteria.

Activities

beach ball

Weather Forecast

spain

Weather for spain: Sunny, 25°C (sample data).

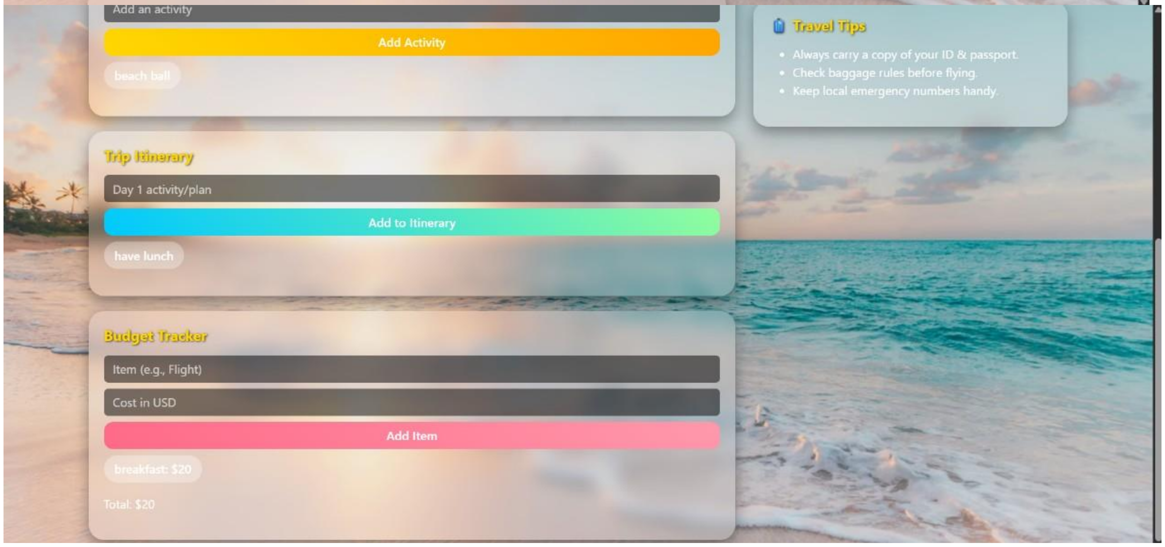
Trip Countdown

26-09-2025

2 days until your trip!

Travel Tips

- Always carry a copy of your ID & passport.
- Check baggage rules before flying.
- Keep local emergency numbers handy.



beach ball

Travel Tips

- Always carry a copy of your ID & passport.
- Check baggage rules before flying.
- Keep local emergency numbers handy.

Trip Itinerary

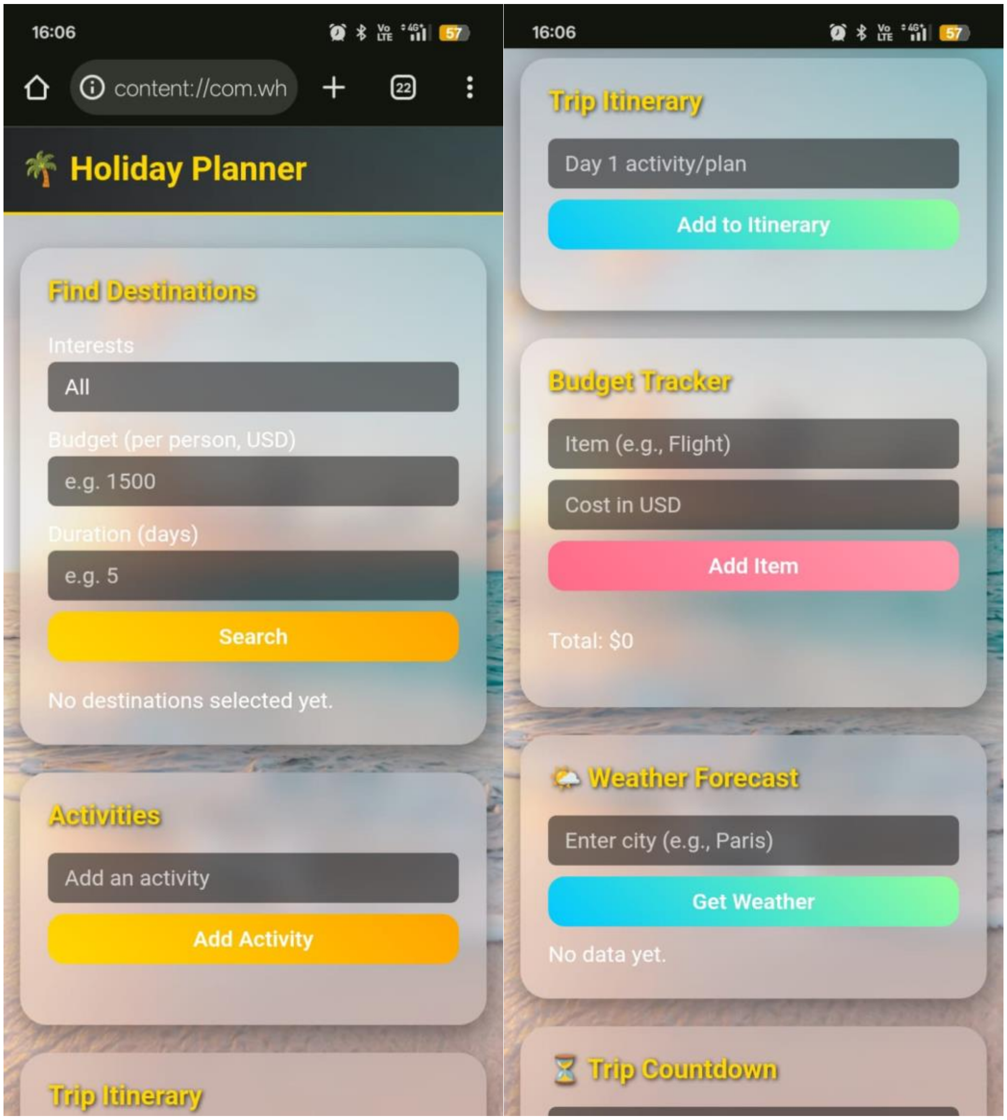
have lunch

Budget Tracker

breakfast: $20

Total: $20

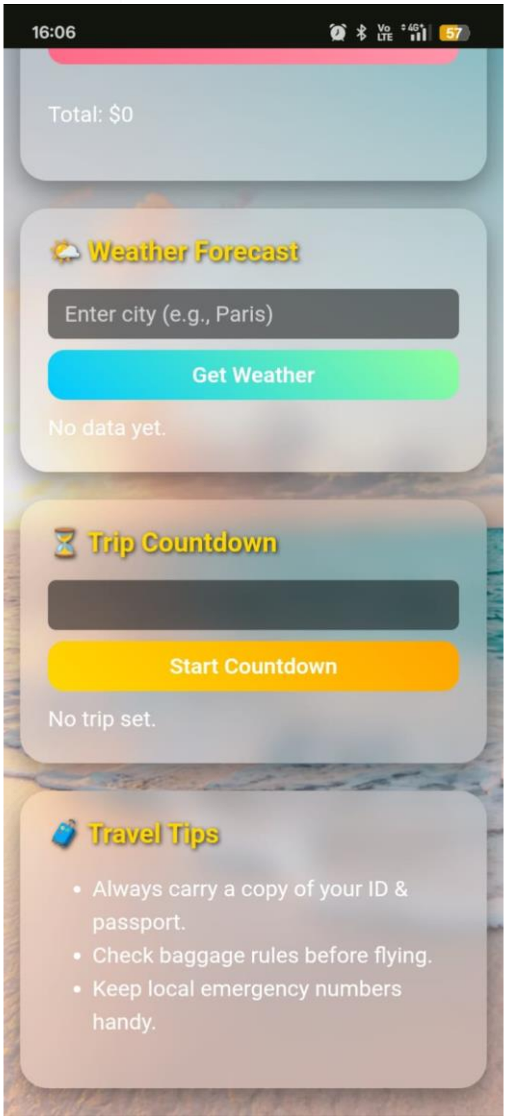
MOBILE VIEW



No destinations selected yet.

Total: $0

No data yet.



Total: $0

🌤️ Weather Forecast

No data yet.

⏳ Trip Countdown

No trip set.

🧳 Travel Tips

- Always carry a copy of your ID & passport.
- Check baggage rules before flying.
- Keep local emergency numbers handy.

11. Conclusion

The Holiday Planner project successfully provides users with an organized and efficient way to plan their vacations. By offering features such as itinerary creation, activity scheduling, and budget management within a clean and responsive interface, the application ensures both functionality and ease of use. Its minimalist design and intuitive navigation make it suitable for users of all ages, while persistent storage guarantees that plans are saved for future access. Overall, the project demonstrates how thoughtful design and simple technology can simplify holiday planning, reduce stress, and enhance the overall travel experience.

12. References

- L&T LMS : https://learn.lntedutech.com/Landing/MyCourse