Firstprogram

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

  <title>Event Management</title>

  <!-- Bootstrap CSS -->

  <link

    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"

    rel="stylesheet"

  />

  <style>

    body {

      font-family: Arial, sans-serif;

    }
    .hero {
      background: #141214;
      color: white;
      padding: 60px 20px;
      text-align: center;
    }
    footer {
      background: #222;
      color: white;
      padding: 15px;
      text-align: center;
    }
  </style>
</head>
<body>
```

```
<!-- HEADER -->
<header>
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container">
      <a class="navbar-brand" href="#">Event Management</a>
      <button
        class="navbar-toggler"
        type="button"
        data-bs-toggle="collapse"
        data-bs-target="#menu"
      >
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="menu">
        <ul class="navbar-nav ms-auto">
          <li class="nav-item"><a class="nav-link" href="#events">More Events</a></li>
      <li class="nav-item"><a class="nav-link" href="#register">Register</a></li>
     </ul>
    </div>
   </div>
  </nav>
 </header>
 <main>
  <section class="hero">
   <h1>Event Management</h1>
  </section>
  <section class="container my-5">
   <div id="eventCarousel" class="carousel slide" data-bs-ride="carousel">
    <div class="carousel-inner">
     <div class="carousel-item active">
      <img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTjgzPMvBrbLPliK-kjJE6Nm8pL_FxvI7_wKA&s" class="d-block w-100" alt="Event 1">
```

```
        </div>

        <div class="carousel-item">

          <img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRFI7ygT65aVwQcrF_RPAqJQvfDXby3GVDyng&s" class="d-block w-100" alt="Event 2">

        </div>

          <div class="carousel-item">

          <img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRPZnM4AdMFiNwD4JODElwtHjPFiDX0WeUXnw&s" class="d-block w-100" alt="Event 2">

        </div>

      </div>

    </div>

  </section>

  <section id="events" class="container my-5">

    <h2 class="text-center mb-4"> Events</h2>

    <div class="row g-4">

      <div class="col-md-4">
  <div class="card h-100">
    <div class="card-body">
      <h5 class="card-title">Tech Conference</h5>
      <p class="card-text">Join with industry experts</p>
      <button class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#detailsModal">
        View
      </button>
    </div>
  </div>
</div>
<div  class="col-md-4">
  <div class="card h-100">
    <div class="card-body">
      <h5 class="card-title">Wedding Event</h5>
      <p class="card-text"> wedding.</p>
```

```
        <button class="btn btn-primary">View </button>

      </div>

    </div>

  </div>

  <div class="col-md-4">

    <div class="card h-100">

      <div class="card-body">

        <h5 class="card-title">Musical Event</h5>

        <p class="card-text"> music</p>

        <button class="btn btn-primary">View </button>

      </div>

    </div>

  </div>

 </div>

</section>

<section id="register" class="container my-5">
```

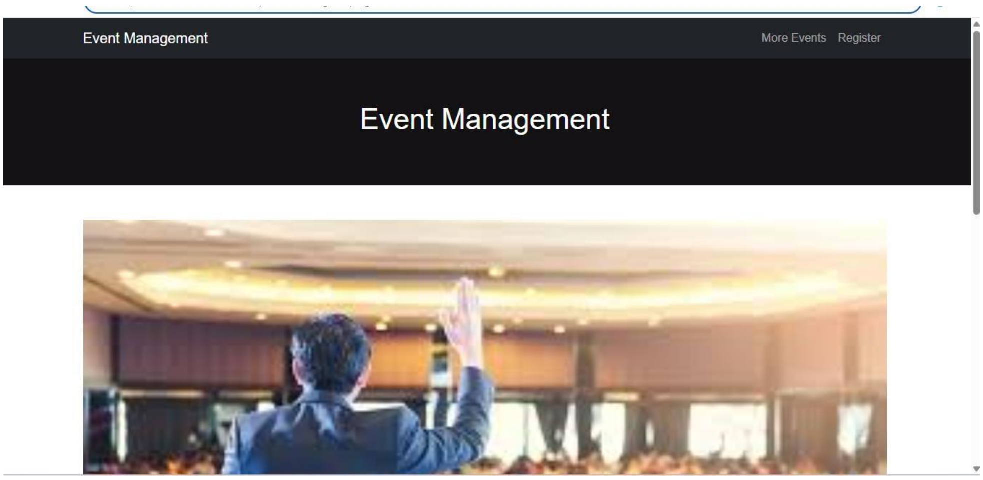
```
<h2 class="text-center mb-4"> Registration</h2>

<form class="row g-3 needs-validation" novalidate>

 <div class="col-md-6">

  <label class="form-label">Name</label>

  <input type="text" class="form-control" required />

  <div class="invalid-feedback">Name is required</div>

 </div>

 <div class="col-md-6">

  <label class="form-label">Email</label>

  <input type="email" class="form-control" required />

  <div class="invalid-feedback">Valid email required</div>

 </div>

 <div class="col-12 text-center">

  <button class="btn btn-success" type="submit">Register</button>

 </div>
```

```html
    </form>

  </section>

</main>

<footer>

  <p>&copy; All Rights Reserved.</p>

</footer>

<div class="modal fade" id="detailsModal" tabindex="-1">

  <div class="modal-dialog">

    <div class="modal-content">

      <div class="modal-header">

        <h5 class="modal-title"> Conference</h5>

        <button class="btn-close" data-bs-dismiss="modal"></button>

      </div>

      <div class="modal-body">

        Learn and upgrade

      </div>
```

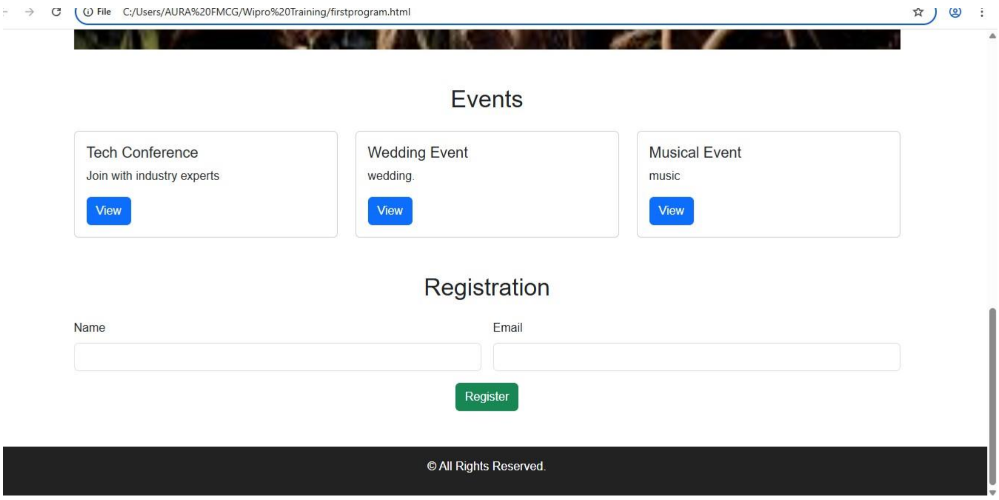
```
   </div>
  </div>
 </div>
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
 <script>
  (() => {
   const forms = document.querySelectorAll(".needs-validation");
   forms.forEach(form => {
    form.addEventListener("submit", event => {
     if (!form.checkValidity()) {
      event.preventDefault();
      event.stopPropagation();
     }
     form.classList.add("was-validated");
    });
```

```
      });

    })();

  </script>

</body>

</html>
```

Output:



Event Management



Events

Tech Conference
Join with industry experts

Wedding Event
wedding.

Musical Event
music

Registration

Name

Email

© All Rights Reserved.

```
}
```

Second program:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <title>Event Dashboard</title>

  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

  <style>

    body {

      font-family: Arial, sans-serif;

      background: #9abee2;
      color: black;

      padding: 20px;
      text-align: center;

    }

    .filters {
```

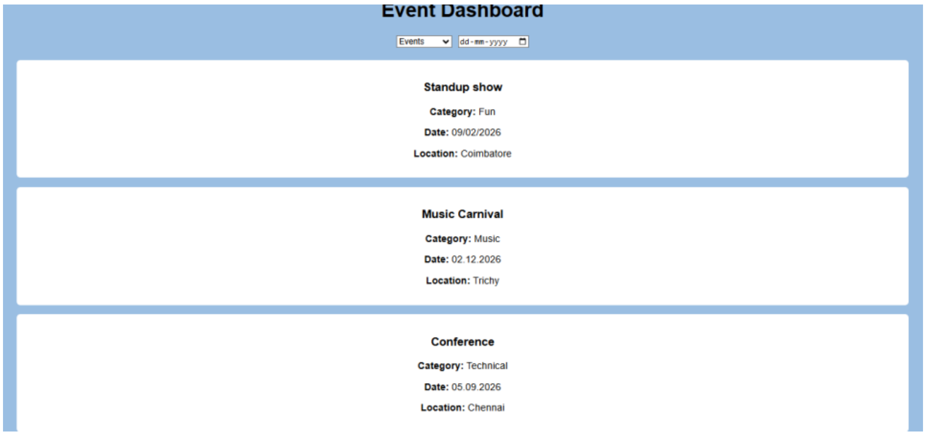
```
    display: flex;

    gap: 10px;

    justify-content: center;

    margin-bottom: 20px;

  }

  .events {

    display: grid;

    gap: 15px;

  }

  .card {

    background: white;

    padding: 15px;

    border-radius: 6px;

  }

</style>
```

```
</head>
<body>
  <h1 style="text-align:center">Event Dashboard</h1>
  <div class="filters">
    <select id="categoryFilter">
      <option value="Events"> Events</option>
      <option value="Fun">Fun</option>
      <option value="Technology">Technology</option>
      <option value="Music">Music</option>

    </select>
    <input type="date" id="dateFilter" />
  </div>
  <div class="events" id="eventContainer"></div>
  <script>
    const Events = [
{
  id: 1,
  title: "Standup show",
  category: "Fun",
  date: "09/02/2026",
  location: "Coimbatore"
},
{
  id: 2,
  title: "Music Carnival",
  category: "Music",
  date: "02.12.2026",
  location: "Trichy"
},
{
    id: 3,

    title: "Conference",

    category: "Technical",

    date: "05.09.2026",

    location: "Chennai"

  }

];

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

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

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

let allEvents = [];

const fetchEvents = async () => {

  return new Promise((resolve, reject) => {

    setTimeout(() => {

      resolve(Events);

    }, 500);
  });
};
const renderEvents = (events) => {
  eventContainer.innerHTML = "";
  events.forEach(({ title, category, date, location }) => {
    eventContainer.innerHTML += `
      <div class="card">
        <h3>${title}</h3>
        <p><strong>Category:</strong> ${category}</p>
        <p><strong>Date:</strong> ${date}</p>
        <p><strong>Location:</strong> ${location}</p>
      </div>
    `;
  });
};
```

```
const applyFilters = () => {
  const selectedCategory = categoryFilter.value;
  const selectedDate = dateFilter.value;
  const filteredEvents = allEvents.filter(event => {
    const categoryMatch =
      selectedCategory === "All" || event.category === selectedCategory;
    const dateMatch =
      !selectedDate || event.date === selectedDate;
    return categoryMatch && dateMatch;
  });
  renderEvents(filteredEvents);
};
categoryFilter.addEventListener("change", applyFilters);
dateFilter.addEventListener("change", applyFilters);
fetchEvents().then(data => {
  allEvents = data;
```

```
   renderEvents(allEvents);

  });

 </script>

</body>

</html>
```

Output:



Event Dashboard

Standup show

Category: Fun

Date: 09/02/2026

Location: Coimbatore

Music Carnival

Category: Music

Date: 02.12.2026

Location: Trichy

Conference

Category: Technical

Date: 05.09.2026

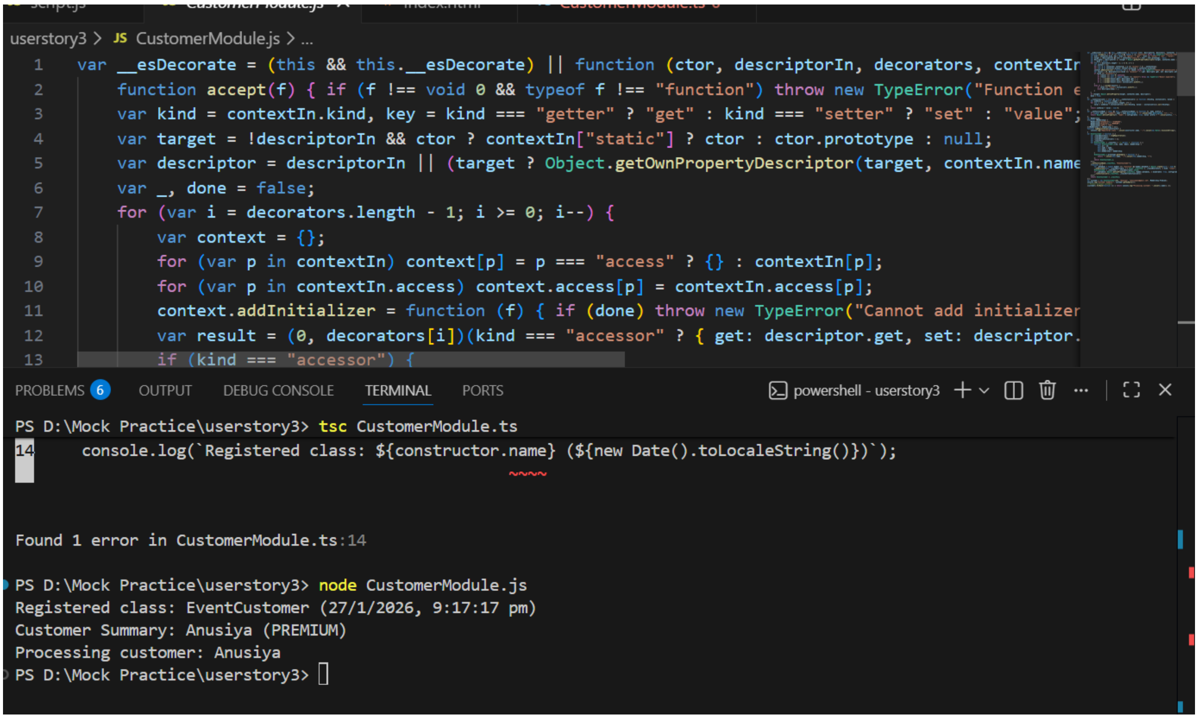
Location: Chennai

Third Program:

```
enum Membership {
  Standard = "STANDARD",
  Premium = "PREMIUM",
  VIP = "VIP",
}

interface Customer {
  id: number;
  name: string;
  email: string;
}
```

```
function logRegistration(constructor: Function) {
  console.log(`Registered class: ${constructor.name} (${new Date().toLocaleString()})`);
}

@logRegistration
class EventCustomer implements Customer {
  constructor(
    public id: number,
    public name: string,
    public email: string,
    public membership: Membership
  ) {}

  getSummary() {
    return `${this.name} (${this.membership})`;
  }
}

const customer = new EventCustomer(101, "Anusiya", "anusiyaanu@gmail.com",
Membership.Premium);

console.log("Customer Summary:", customer.getSummary());

// Process multiple customers
[customer].forEach(c => console.log(`Processing customer: ${c.name}`));
```

Output:



```
var __esDecorate = (this && this.__esDecorate) || function (ctor, descriptorIn, decorators, contextIr
    function accept(f) { if (f !== void 0 && typeof f !== "function") throw new TypeError("Function e
    var kind = contextIn.kind, key = kind === "getter" ? "get" : kind === "setter" ? "set" : "value";
    var target = !descriptorIn && ctor ? contextIn["static"] ? ctor : ctor.prototype : null;
    var descriptor = descriptorIn || (target ? Object.getOwnPropertyDescriptor(target, contextIn.name
    var _, done = false;
    for (var i = decorators.length - 1; i >= 0; i--) {
        var context = {};
        for (var p in contextIn) context[p] = p === "access" ? {} : contextIn[p];
        for (var p in contextIn.access) context.access[p] = contextIn.access[p];
        context.addInitializer = function (f) { if (done) throw new TypeError("Cannot add initializer
        var result = (0, decorators[i])(kind === "accessor" ? { get: descriptor.get, set: descriptor.
        if (kind === "accessor") {
```

```
PS D:\Mock Practice\userstory3> tsc CustomerModule.ts
14     console.log(`Registered class: ${constructor.name} (${new Date().toLocaleString()})`);
                                                            ~~~~


Found 1 error in CustomerModule.ts:14

PS D:\Mock Practice\userstory3> node CustomerModule.js
Registered class: EventCustomer (27/1/2026, 9:17:17 pm)
Customer Summary: Anusiya (PREMIUM)
Processing customer: Anusiya
PS D:\Mock Practice\userstory3> 
```