Training Project Lab documentation

presented by: Balkis Karoui

supervisor: Viktória Nemkin Dr

Introduction

EventNest is a dynamic and user-centric event management platform, designed with a modern 3-tier architecture. This architecture divides the application into three distinct layers: the Presentation Layer (Frontend), the Business Logic Layer (Backend), and the Data Layer (Database).

Our documentation provides an in-depth exploration of each layer, focusing on how they seamlessly integrate to create a robust, scalable, and efficient application. The Frontend is built using Angular, offering an interactive and intuitive user interface. The Backend, powered by Node.js and Express, handles business logic, authentication, and server-side processing. The Data Layer, utilizing MongoDB, manages data storage and retrieval, ensuring data integrity and performance.

Throughout this documentation, we will delve into the specifics of each component, service, and model, elucidating their roles and interactions within the system. Our goal is to present a clear understanding of EventNest's architecture, highlighting its strengths and the advanced technologies it employs.

Data Layer Properties

The EventNest backend architecture is designed to manage event-related data and user interactions effectively. The application leverages Node.js with Express for server-side operations and MongoDB as its database. This documentation will focus on the data layer properties of the backend, emphasizing models, routes, and data handling logic.

Models

The backbone of the EventNest data layer consists of two primary models: `Event` and `User`.

1. Event Model

- Schema: Includes fields like `Title`, `Category`, `Date`, `Time`, `location`, `description`, and `imageUrl`. The `Date` field is particularly handled as a string for display purposes, though it's tested and processed as a `Date` object in CRUD operations.
- Collection: Stored in the 'Events' collection in MongoDB.
- Usage: Represents event data within the application, crucial for all event-related operations.

2. User Model

- Schema: Consists of `Username`, `email`, `password`, `rsvps`, and `myEvents`. The `rsvps` and `myEvents` fields are arrays referencing the `Event` model, indicating the events that a user has RSVPed to or created.
- Collection: Part of the 'Users' collection in MongoDB.
- Usage: Central to handling user data, including authentication, event RSVPs, and user-created events.

Routing and Controllers

The application's routing is handled through Express.js, with specific routes dedicated to different functionalities.

Event Routes

- Add Event: `router.post('/add')` allows for creating new events, including handling image uploads using `multer`.
- List Events: `router.get('/all')` retrieves all events.
- Find Event by ID: `router.get('/find/:id')` fetches a single event by its ID.
- Find Events by Category: `router.get('/byCategory/:category')` filters events by category.
- Update Event: `router.put('/update/:id')` updates event details.
- Delete Event: `router.delete('/delete/:id')` removes an event.

User Routes

- User Registration: `router.post('/register')` includes password hashing using `bcrypt`.
- User Login: `router.post('/login')` authenticates users, checking hashed passwords.
- RSVP Handling: Routes to add and retrieve RSVP'd events for users.
- User Events: Routes to manage and retrieve events created by a user.

Key Features and Logic

1. Image Upload

- Implemented using `multer` for handling file uploads, specifically configured to accept and store image files.

2. Password Security

- Utilizes `bcrypt` for hashing user passwords, enhancing security.

3. Data Validation and Error Handling

- Each route includes error handling and validation logic, ensuring robust and reliable server responses.

4. Relational Data Handling

- `User` model's `rsvps` and `myEvents` fields create relations with the `Event` model, demonstrating MongoDB's capability to handle relational data.

5. Populating Query Results

- The use of `.populate()` in routes like `router.get('/user/:userId/rsvp')` showcases the ability to join data from different collections, similar to SQL JOIN operations.

Server Setup and Middleware Configuration

In addition to the previously documented components (models, routes, and controllers), the EventNest backend includes a server setup and middleware configuration, crucial for overall application functionality. This section extends the documentation to cover these aspects.

Server Initialization and Configuration

Express.js Application

- Initialization: The backend uses Express.js, a fast, unopinionated, minimalist web framework for Node.js, to create the server.
- Usage: Express.js facilitates the creation of an HTTP server and is used to define request endpoints, request/response handling, and middleware.

CORS Middleware

- Purpose: Cross-Origin Resource Sharing (CORS) middleware is used to enable cross-origin requests, essential for a frontend application (running on a different domain) to interact with this backend.

- Configuration: The `cors()` middleware is applied globally to allow requests from all origins, ensuring seamless frontend-backend communication.

Database Connection

- Module: The `require('./config/connect')` line indicates the existence of a separate module to handle database connection logic, likely connecting to a MongoDB database.

Routing Configuration

Event and User Routes

- EventRoute: Managed by the `EventRoute` module, handling all event-related endpoints.
- UserRoute: Managed by the `UserRoute` module, encompassing user authentication, registration, and user-specific operations.

Endpoint Prefix

- URL Structure: Routes are prefixed with `/TPL/Events` for event-related operations and `/TPL/Users` for user-related operations, indicating clear, structured URL design for API endpoints.

Static File Serving

- Purpose: Serves static files from the 'uploads' directory, enabling access to uploaded images.
- Implementation: The `app.use('/uploads', express.static('uploads'))` line makes the files in the 'uploads' directory accessible via HTTP requests.

Error Handling Middleware

- Functionality: A generic error-handling middleware captures and responds to any unhandled errors in the application.
- Response: In case of an error, it logs the error stack and sends a `500 Internal Server Error` response.

Server Listening

- Port Configuration: The server listens on port **3000**, a common default port for local development in Node.js applications.

User interface Layer properties

EventNest's Angular frontend is a sophisticated, user-friendly interface designed for managing and participating in events. This documentation provides an overview of the key components and services, as well as their functionalities and interactions within the application.

Component Breakdown

1. CreateEventComponent

- Purpose: Allows users to create new events.
- Features:
    - Form fields for event details (`EventData` interface).
    - Image upload functionality with file handling (`onFileSelected` method).
    - Form submission (`onCreateEvent` method) with validation to ensure user is logged in.
    - Integration with `EventsAPIService` for backend communication and `AuthService` for user authentication.
    - Redirection using `Router` upon successful event creation or if user authentication fails.

2. FeaturedEventsComponent

- Purpose: Displays a list of featured events.
- Features:
    - Fetches and displays event data using `EventsAPIService`.
    - Implements RSVP functionality with `promptRsvp` and `rsvpToEvent` methods.
    - Utilizes `SharedService` for event updates and `AuthService` for user authentication.

3. FilterNavBarComponent

- Purpose: Provides a filtering interface for events.
- Features:
    - Predefined event categories for filtering.
    - Event filtering logic with `filterEvents` method.
    - Communicates with `EventsAPIService` for fetching events and `SharedService` for updating the event list.

4. HeaderComponent

- Purpose: Manages the application's navigation header.
- Features:
    - Dynamically displays navigation options based on user login status using `AuthService`.
    - Implements a dropdown menu for user-specific options.
    - Logout functionality that redirects to the login page.

5. UsersMyEventsComponent

- Purpose: Displays events created by the logged-in user.
- Features:
    - Fetches user-specific events using `EventsAPIService`.
    - Deletes user events with confirmation dialog using `Swal` (SweetAlert).

6.Create Event Component

is an Angular component in the EventNest application responsible for facilitating the creation of new events. This component encompasses user input handling, form data preparation, and integration with services for backend communication.

Interface: EventData

- Defines the structure of the event data with properties like `eventName`, `eventDate`, `eventTime`, `eventPlace`, `eventCategory`, `eventDescription`, and `eventImage`.
- `eventImage` is of type `File | null`, which allows for handling file uploads.

Constructor

- The component's constructor injects three services:
    - `EventsAPIService` for backend communication related to event management.
    - `AuthService` for user authentication checks.
    - `Router` for navigation control.

File Handling Method: onFileSelected

- Triggered when a user selects a file (image) to upload.
- Sets `eventData.eventImage` to the selected file.

Event Creation Method: onCreateEvent

- Checks if the user is logged in using `AuthService`. If not, redirects to the login page.
- Retrieves the current user ID for associating the created event with the user.
- Prepares a `FormData` object with event details and the selected image file.
- Calls `EventsAPIService.createEvent` to send the data to the backend.
- Upon successful event creation, adds the event to the user's "My Events" list and navigates to the "My Events" page.
- Includes error handling for both event creation and adding to "My Events".

Routing and Navigation

AppRoutingModule

- Defines routes and their corresponding components.
- Implements route guarding with `AuthGuardService` for protected routes like event creation.
- Includes nested routes within `LayoutComponent` for a consistent layout across different views.

Business Layer properties: Services

EventsAPIService

- Centralizes communication with the backend for event-related operations.
- **Purpose**: Centralizes all HTTP requests related to events.
- **Testing**: Covered by unit tests to ensure its proper creation and functionality.

AuthService

- Manages user authentication and provides utility methods for checking login status and retrieving the current user ID.
- Purpose: Manages user authentication, session handling, and user data retrieval.
- Key Features:
    - `loginStatusSubject` and `loginStatus$`: Manages and broadcasts the user's login status.
    - `register`, `login`, `logout`: Handle user registration, login, and logout processes.
    - `setUser` and `clearUser`: Manage the current user's session data in sessionStorage.
    - `getCurrentUserId` and `getUserInitials`: Retrieve user-related information from the session.
    - `reinitializeUser`: Re-establishes user data from sessionStorage, typically used on application load.
    - Ensures the persistence of authentication state across browser sessions.

we ensured Session hold Handling in Non-Browser Environments: Checks for `window` object to ensure that session handling code runs only in a browser environment, which is important for universal or server-side rendered applications.

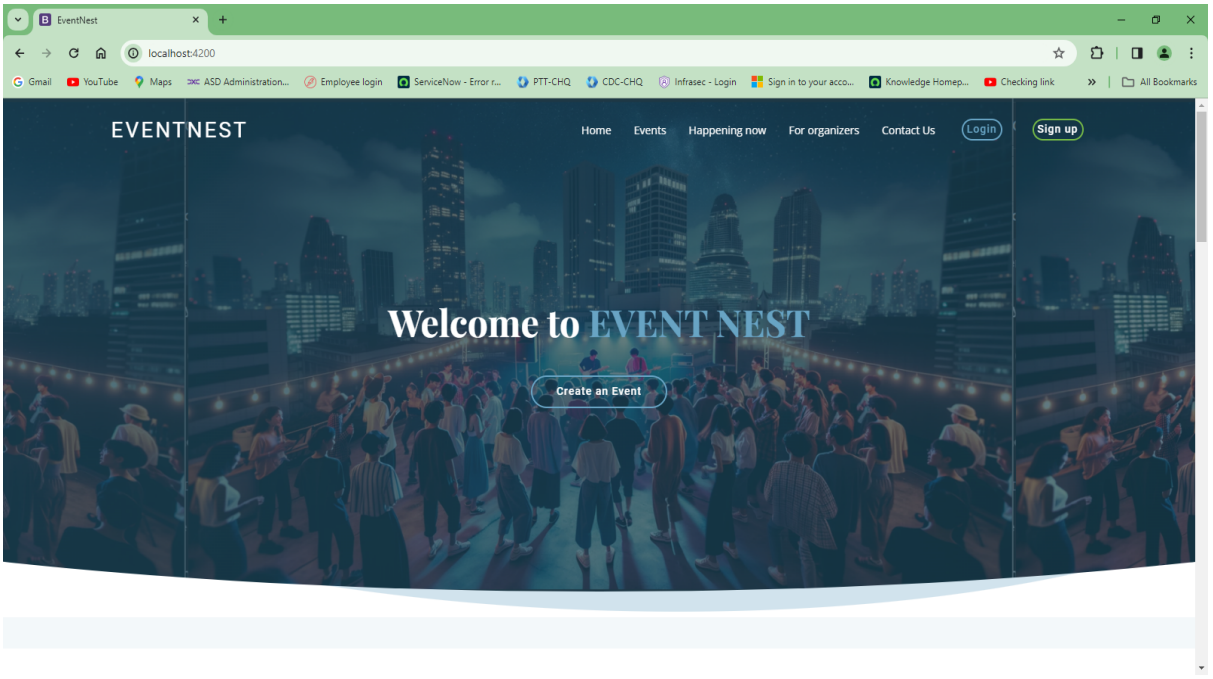
SharedService

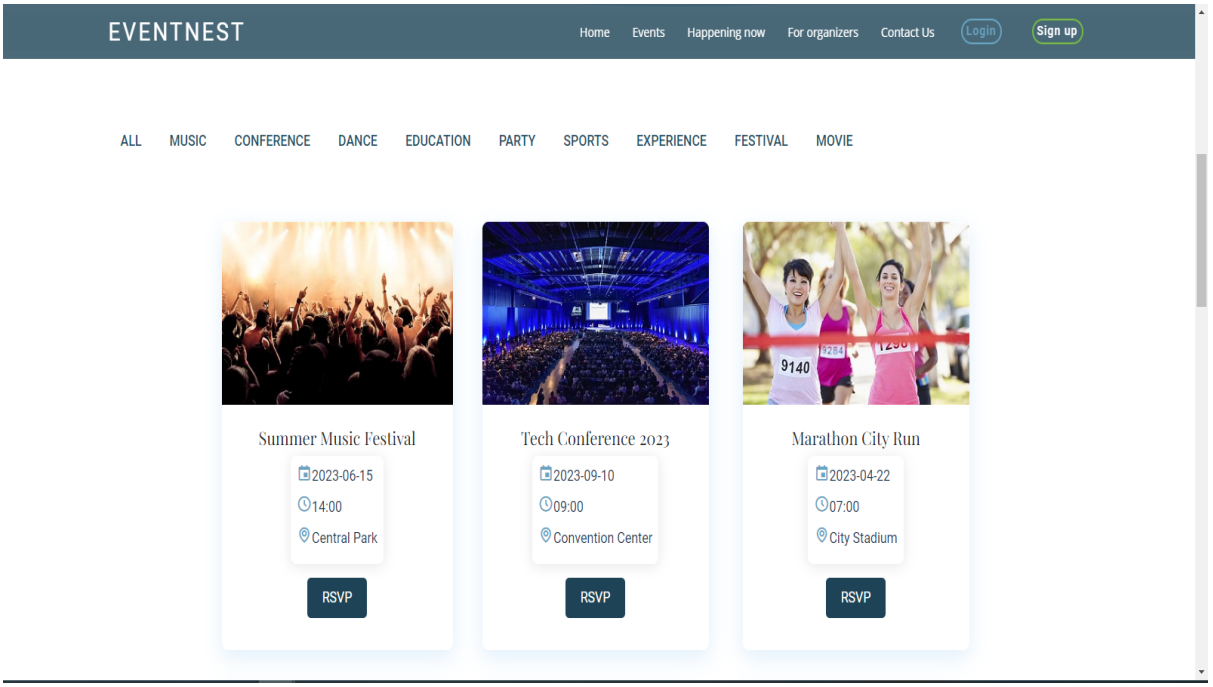
- Facilitates cross-component communication, primarily used for updating event lists in response to user actions.
- Purpose: Facilitates communication and data sharing between components.
- Key Features:
    - Utilizes a `BehaviorSubject` to maintain a list of events (`eventsSource`).
    - `updateEvents` method allows updating this list, which components can subscribe to via `events$`.
    - Ensures real-time updates across components when the event list changes.
- allows components to reactively update whenever the event list changes, demonstrating an effective use of RxJS for state management.

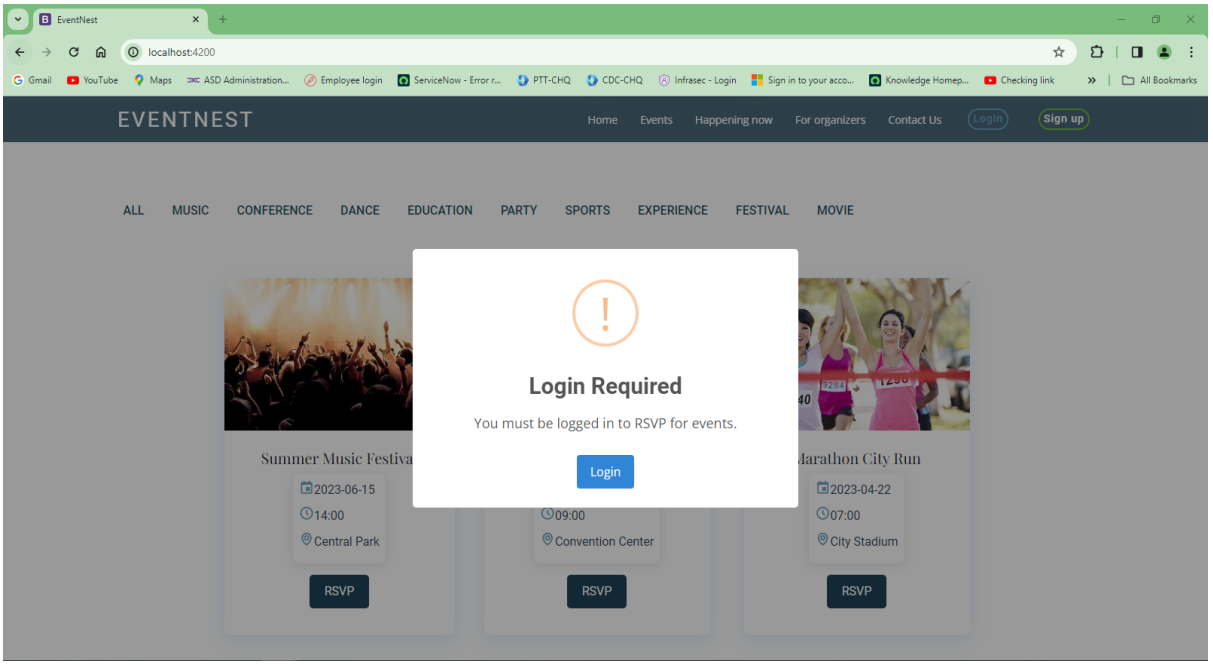
AuthGuardService

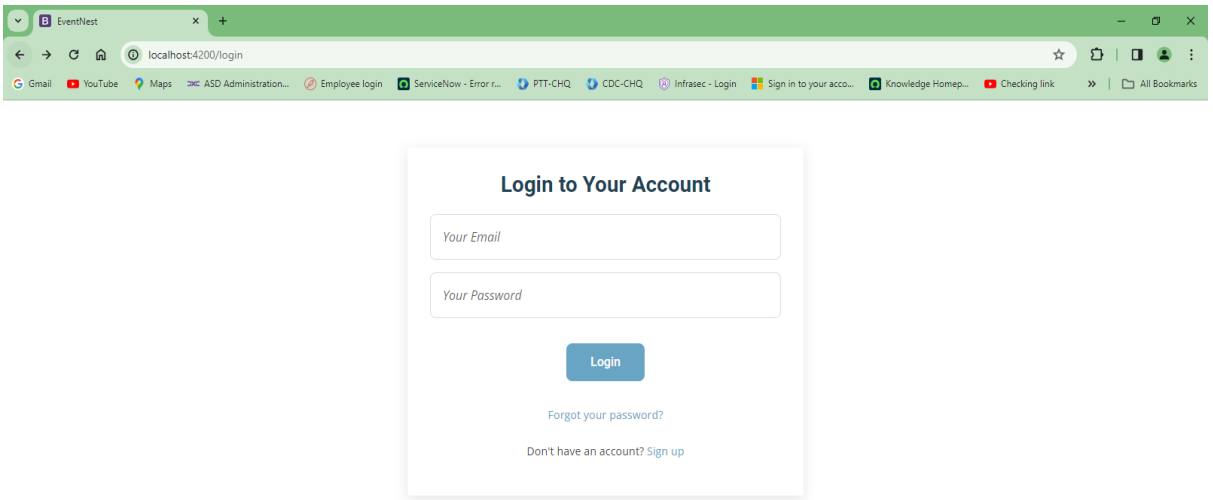
- Route Guarding Logic: The `canActivate` method showcases a typical implementation of route guarding in Angular, using a service to check user authentication and redirecting accordingly.

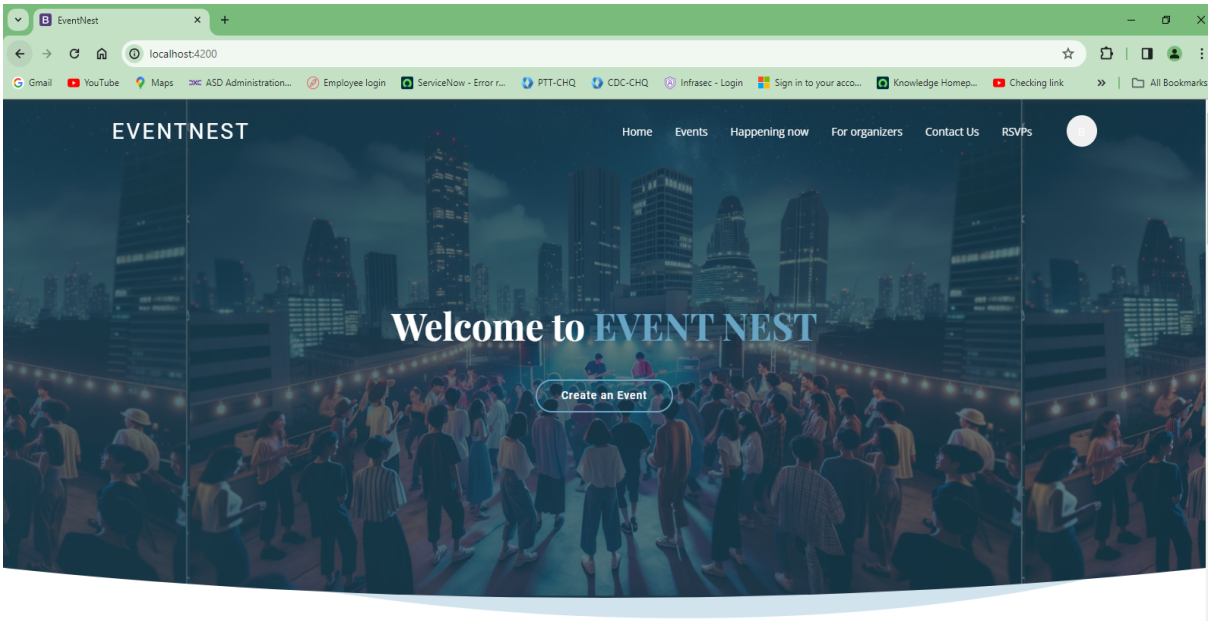
Features and Functionalities:

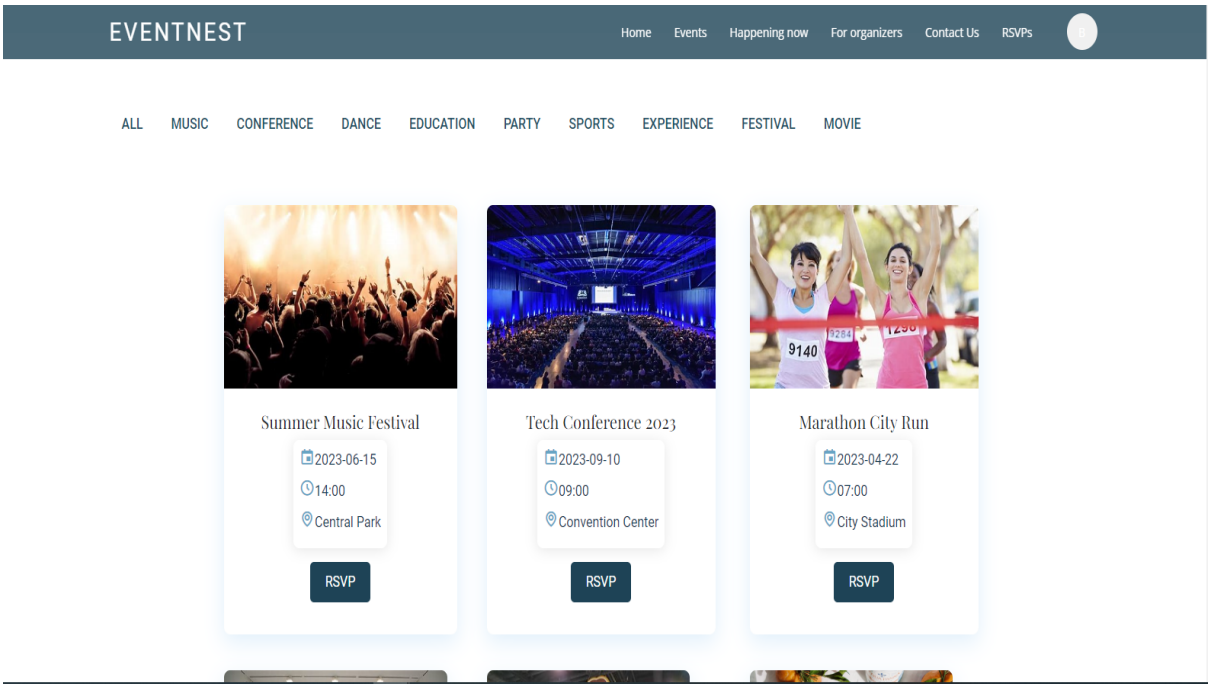


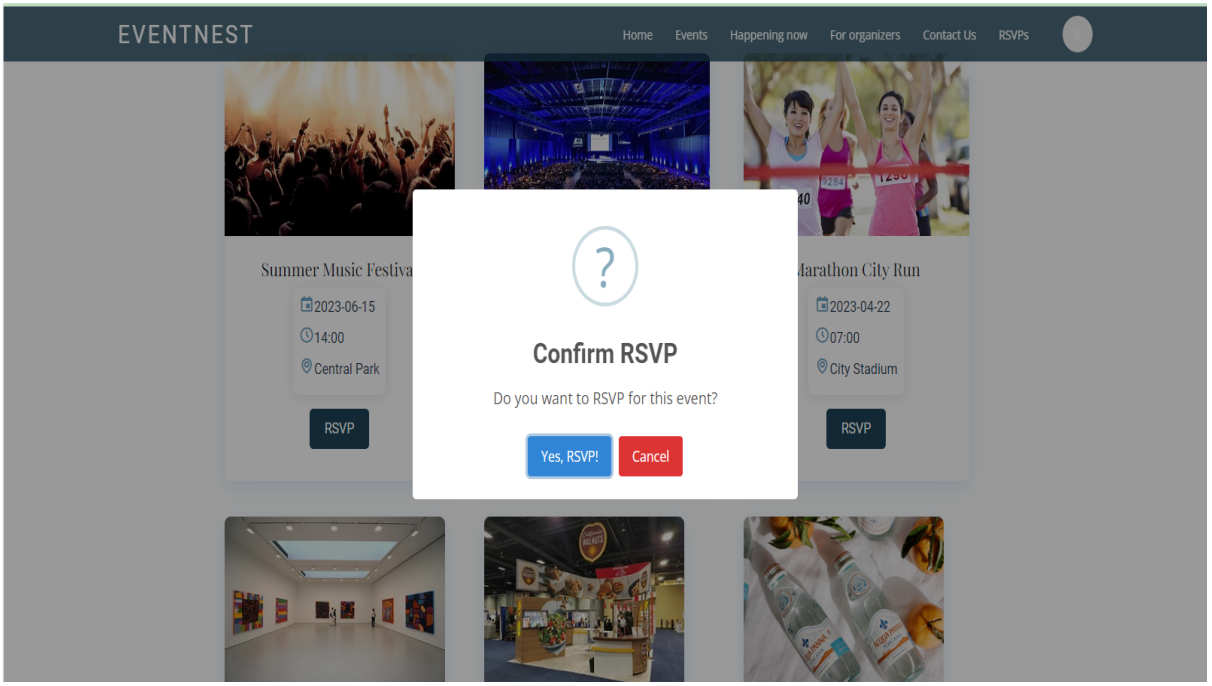



EventNest
localhost:4200
EVENTNEST
Home
Events
Happening now
For organizers
Contact Us
Login
Sign up
ALL
MUSIC
CONFERENCE
DANCE
EDUCATION
PARTY
SPORTS
EXPERIENCE
FESTIVAL
MOVIE
Login Required
You must be logged in to RSVP for events.
Login
2023-06-15
14:00
Central Park
RSVP
09:00
Convention Center
RSVP
2023-04-22
07:00
City Stadium
RSVP

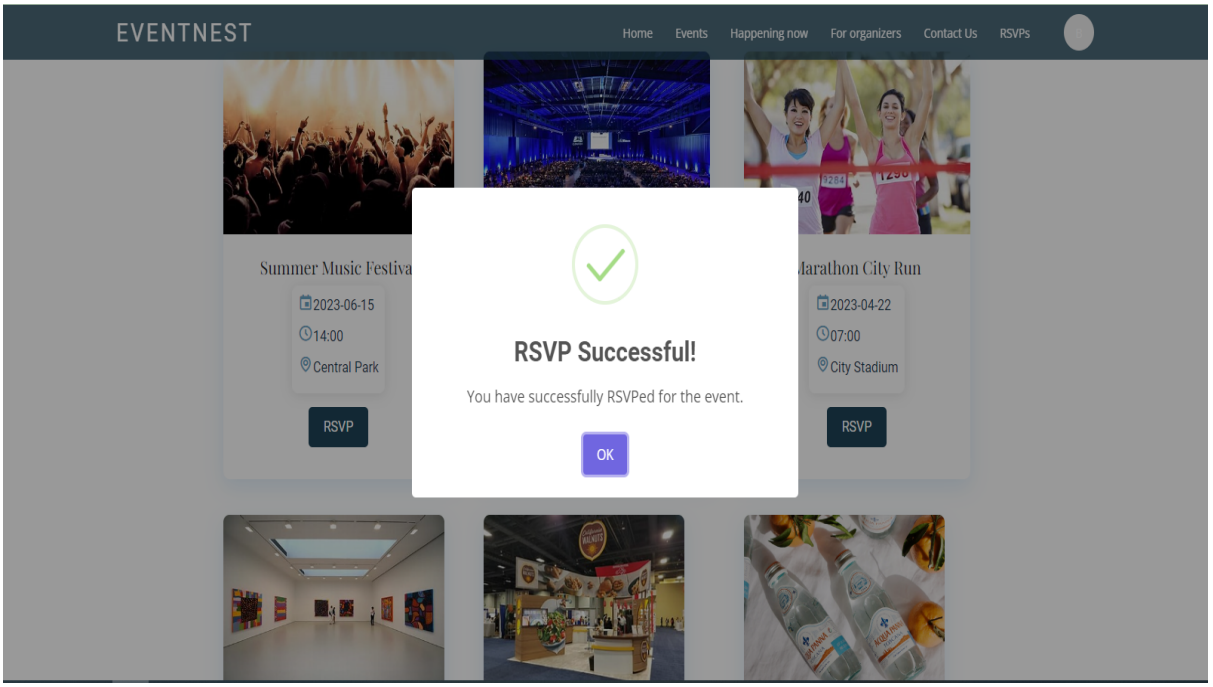

EventNest
localhost:4200/login
Login to Your Account
Your Email
Your Password
Login
Forgot your password?
Don't have an account? Sign up

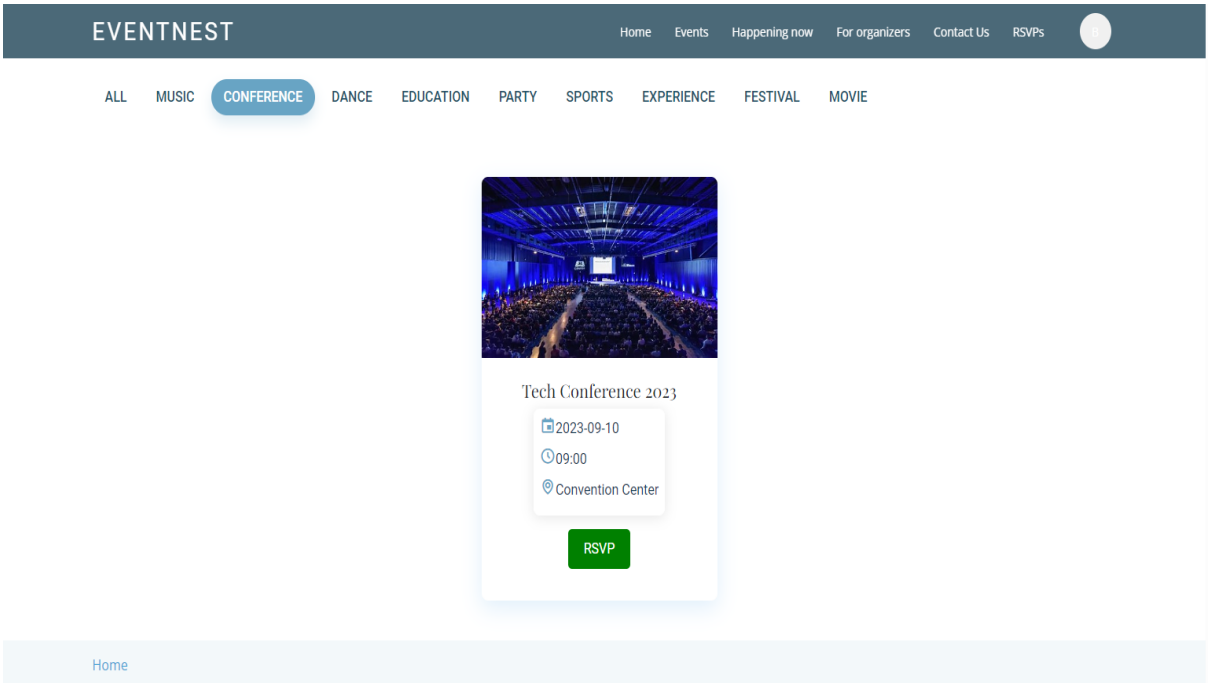

EventNest
localhost:4200
Gmail
YouTube
Maps
ASD Administration...
Employee login
ServiceNow - Error r...
PTT-CHQ
CDC-CHQ
Infrasec - Login
Sign in to your acco...
Knowledge Homep...
Checking link
All Bookmarks
EVENTNEST
Home
Events
Happening now
For organizers
Contact Us
RSVPs
Welcome to EVENT NEST
Create an Event

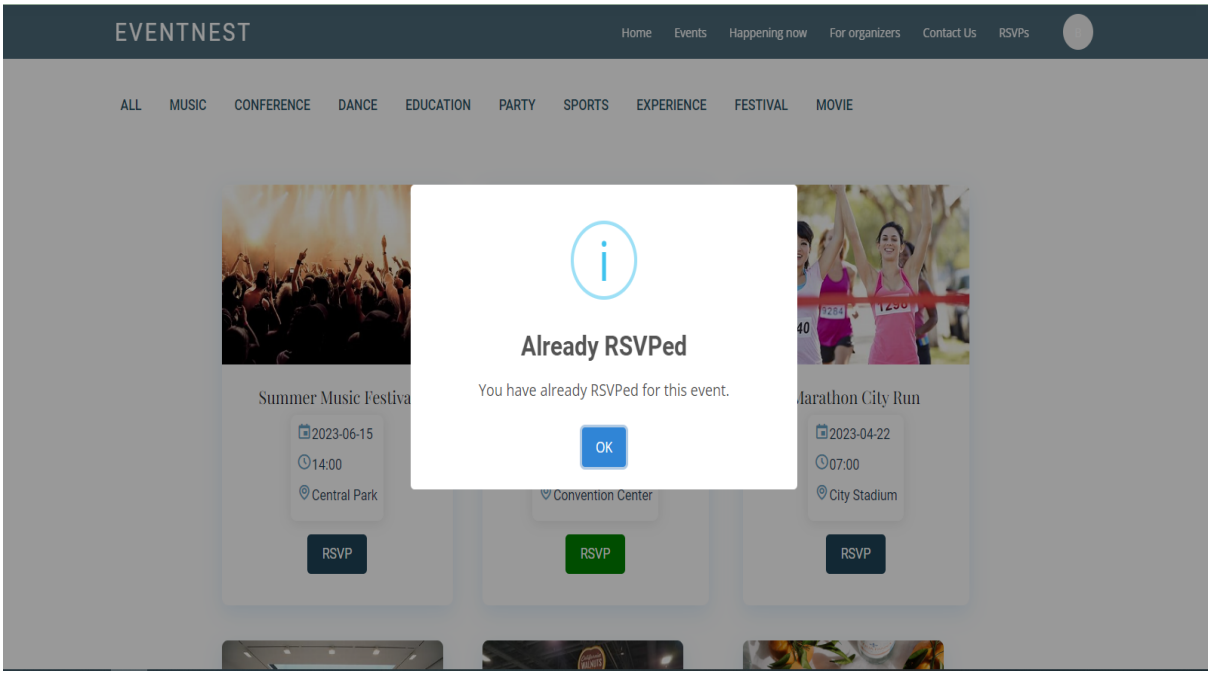

EVENTNEST
Home
Events
Happening now
For organizers
Contact Us
RSVPs
ALL
MUSIC
CONFERENCE
DANCE
EDUCATION
PARTY
SPORTS
EXPERIENCE
FESTIVAL
MOVIE
Summer Music Festival
2023-06-15
14:00
Central Park
RSVP
Tech Conference 2023
2023-09-10
09:00
Convention Center
RSVP
Marathon City Run
2023-04-22
07:00
City Stadium
RSVP

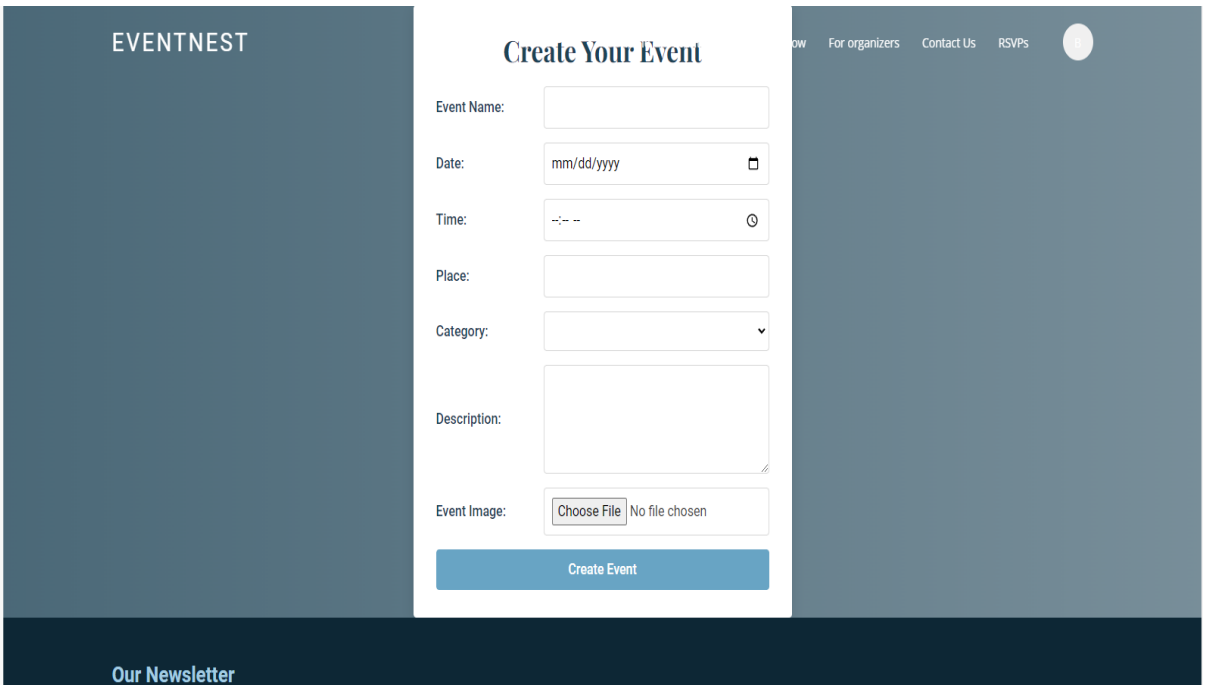

EVENTNEST
Home
Events
Happening now
For organizers
Contact Us
RSVPs
Summer Music Festiva
2023-06-15
14:00
Central Park
RSVP
Confirm RSVP
Do you want to RSVP for this event?
Yes, RSVP!
Cancel
Marathon City Run
2023-04-22
07:00
City Stadium
RSVP

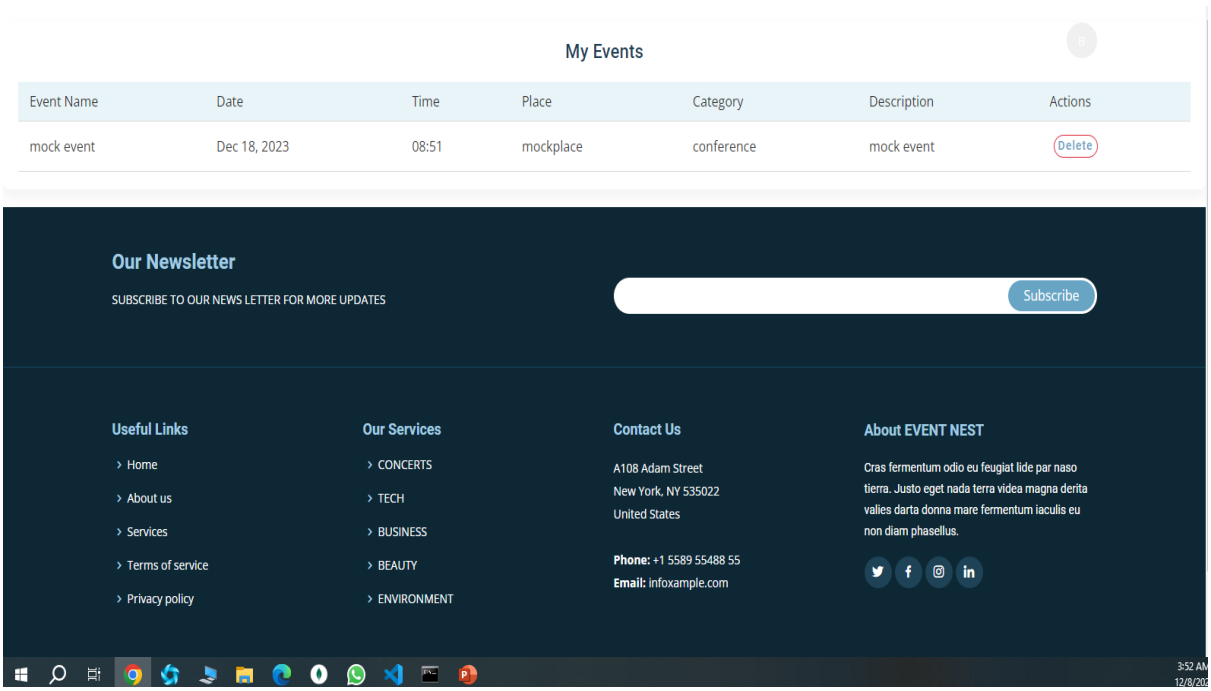
,

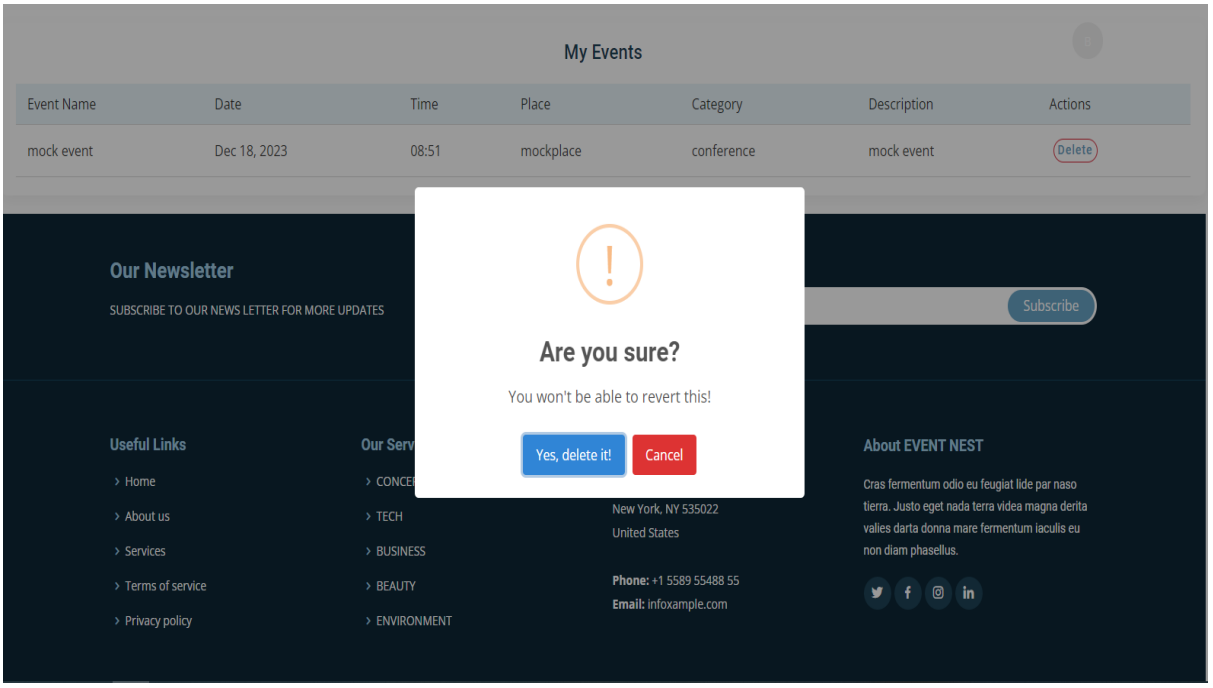

EVENTNEST
Home
Events
Happening now
For organizers
Contact Us
RSVPs
Summer Music Festiva
2023-06-15
14:00
Central Park
RSVP
RSVP Successful!
You have successfully RSVPed for the event.
OK
Marathon City Run
2023-04-22
07:00
City Stadium
RSVP

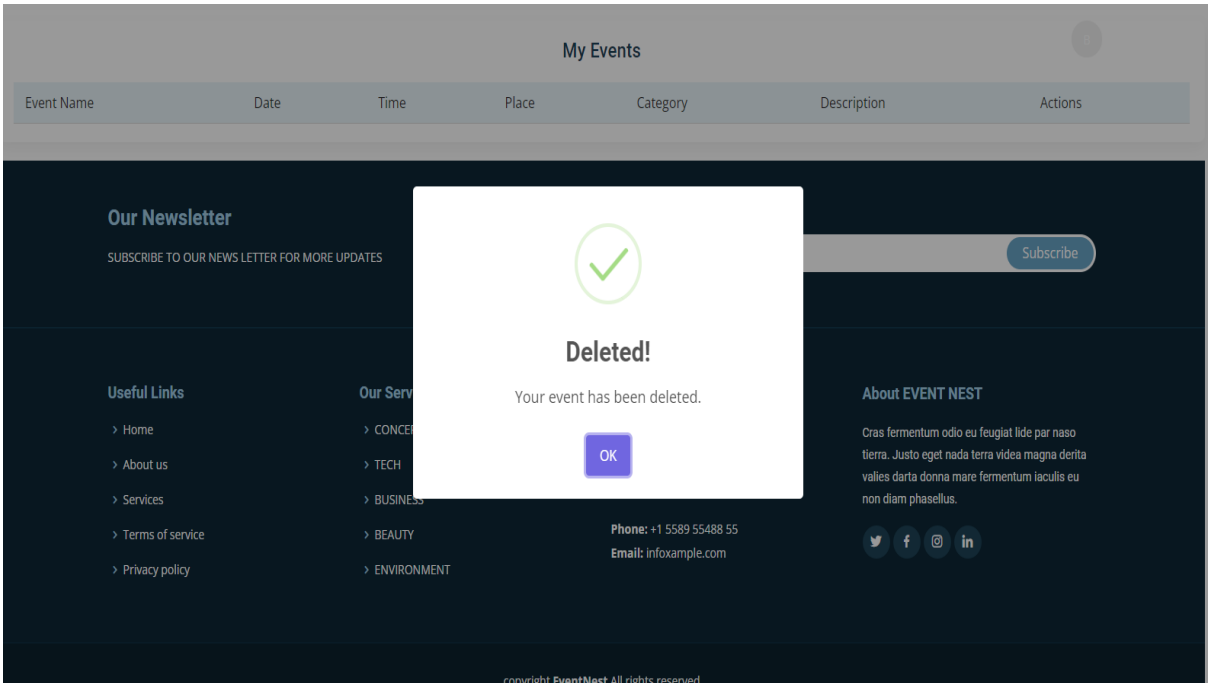

EVENTNEST
Home
Events
Happening now
For organizers
Contact Us
RSVPs
ALL
MUSIC
CONFERENCE
DANCE
EDUCATION
PARTY
SPORTS
EXPERIENCE
FESTIVAL
MOVIE
Tech Conference 2023
2023-09-10
09:00
Convention Center
RSVP
Home

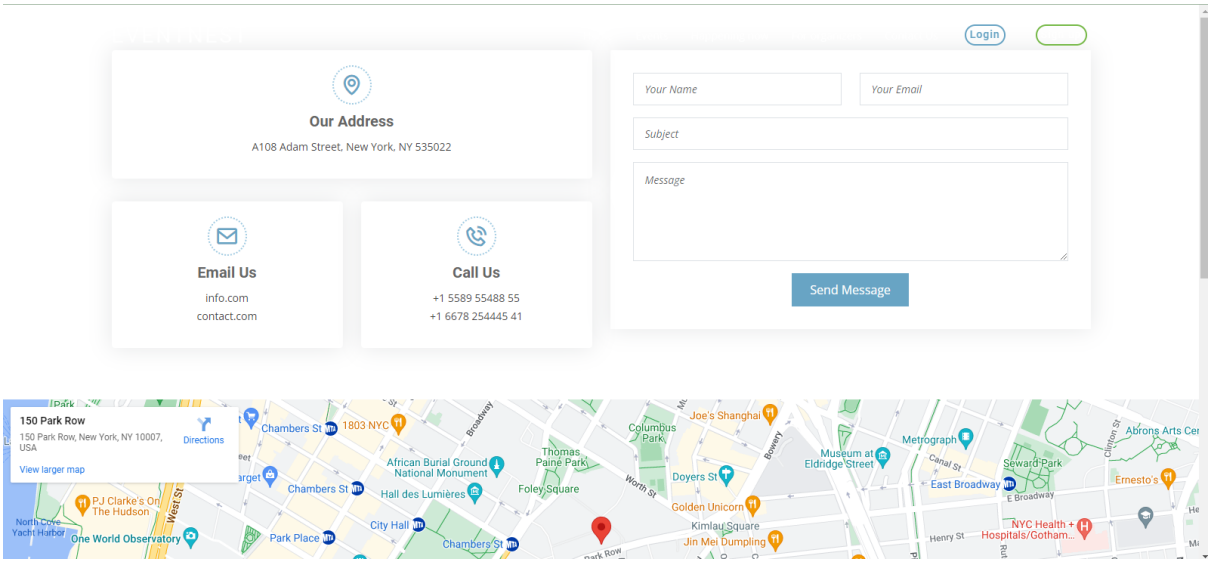

EVENTNEST
Home
Events
Happening now
For organizers
Contact Us
RSVPs
ALL
MUSIC
CONFERENCE
DANCE
EDUCATION
PARTY
SPORTS
EXPERIENCE
FESTIVAL
MOVIE
Already RSVPed
You have already RSVPed for this event.
OK
Summer Music Festiva
2023-06-15
14:00
Central Park
RSVP
Convention Center
RSVP
Marathon City Run
2023-04-22
07:00
City Stadium
RSVP

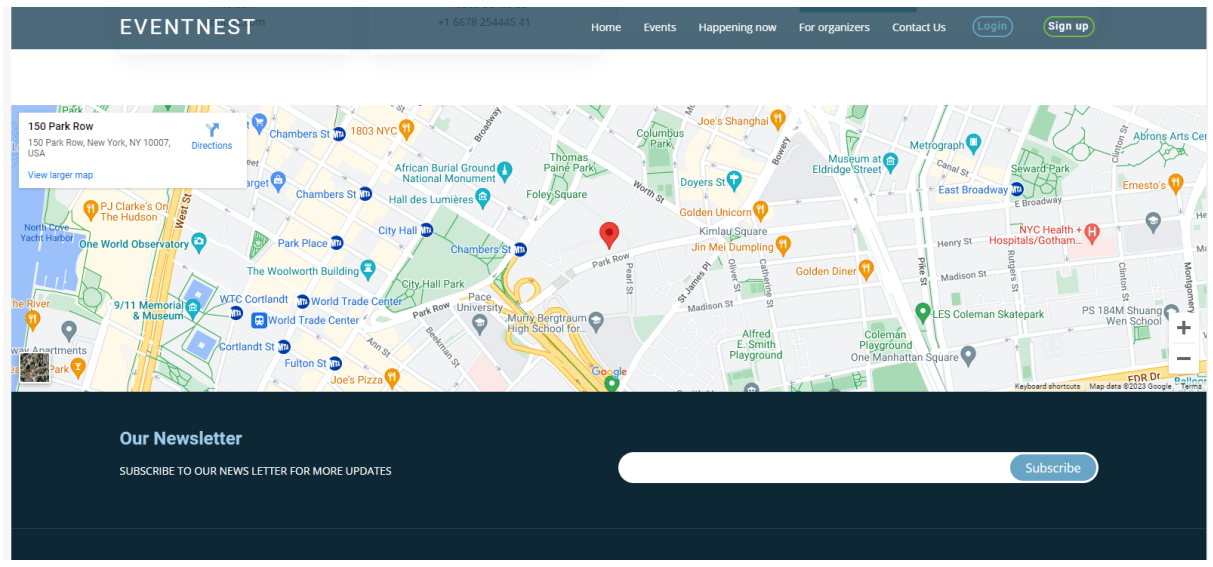

EVENTNEST
For organizers
Contact Us
RSVPs
Create Your Event
Event Name:
Date:
mm/dd/yyyy
Time:
--:-- --
Place:
Category:
Description:
Event Image:
Choose File
No file chosen
Create Event
Our Newsletter


My Events
Event Name
Date
Time
Place
Category
Description
Actions
mock event
Dec 18, 2023
08:51
mockplace
conference
mock event
Delete
Our Newsletter
SUBSCRIBE TO OUR NEWS LETTER FOR MORE UPDATES
Subscribe
Useful Links
Home
About us
Services
Terms of service
Privacy policy
Our Services
CONCERTS
TECH
BUSINESS
BEAUTY
ENVIRONMENT
Contact Us
A108 Adam Street
New York, NY 535022
United States
Phone: +1 5589 55488 55
Email: infoxample.com
About EVENT NEST
Cras fermentum odio eu feugiat lide par naso tierra. Justo eget nada terra videa magna derita valies darta donna mare fermentum iaculis eu non diam phasellus.


My Events
Event Name
Date
Time
Place
Category
Description
Actions
mock event
Dec 18, 2023
08:51
mockplace
conference
mock event
Delete
Are you sure?
You won't be able to revert this!
Yes, delete it!
Cancel
Our Newsletter
SUBSCRIBE TO OUR NEWS LETTER FOR MORE UPDATES
Subscribe
Useful Links
Home
About us
Services
Terms of service
Privacy policy
TECH
BUSINESS
BEAUTY
ENVIRONMENT
New York, NY 535022
United States
Phone: +1 5589 55488 55
Email: infoxample.com
About EVENT NEST
Cras fermentum odio eu feugiat lide par naso tierra. Justo eget nada terra videa magna derita valies darta donna mare fermentum iaculis eu non diam phasellus.


My Events
Event Name
Date
Time
Place
Category
Description
Actions
Our Newsletter
SUBSCRIBE TO OUR NEWS LETTER FOR MORE UPDATES
Subscribe
Deleted!
Your event has been deleted.
OK
Useful Links
Home
About us
Services
Terms of service
Privacy policy
TECH
BEAUTY
ENVIRONMENT
Phone: +1 5589 55488 55
Email: infoxample.com
About EVENT NEST
Cras fermentum odio eu feugiat lide par naso tierra. Justo eget nada terra videa magna derita valies darta donna mare fermentum iaculis eu non diam phasellus.


Login
Our Address
A108 Adam Street, New York, NY 535022
Email Us
info.com
contact.com
Call Us
+1 5589 55488 55
+1 6678 254445 41
Your Name
Your Email
Subject
Message
Send Message
150 Park Row
150 Park Row, New York, NY 10007, USA
Directions
View larger map


EVENTNEST
+1 6678 254445 41
Home
Events
Happening now
For organizers
Contact Us
Login
Sign up
150 Park Row
150 Park Row, New York, NY 10007, USA
Directions
View larger map
Our Newsletter
SUBSCRIBE TO OUR NEWS LETTER FOR MORE UPDATES
Subscribe

Conclusion

In conclusion, the EventNest application stands as a testament to the effectiveness of the 3-tier architecture in building full-fledged web applications. Through the separation of concerns across the three layers, EventNest achieves high levels of maintainability, scalability, and performance.