Enhancing Event Exploration and Engagement: A Social Events Networking Platform Leveraging Cosine Similarity Recommendations and Google Maps Integration

Phul Babu Jha, Bidur Nepal, Anjay Kumar Mishra, Naseeb Dangi and Prajwal Bhandari

1Head, School of Science and Technology, Madan Bhandari Memorial College
2Asst. Prof. Patan Multiple Campus, Tribhuvan University
3PhD & Research Professor, Srinivas University, India
4BCA Scholar, Madan Bhandari Memorial College
5Institute of Science and Technology, Tribhuvan University

ARTICLE INFO

Corresponding Author
Phul Babu Jha

Email
head_schoolsot@mbmc.edu.np

Article History
Received: 12 February 2024
Accepted: 23 March 2024

Orcid
https://orcid.org/0009-0001-3645-5678

Cite

ABSTRACT

This project introduces a Social Events Networking Platform, strategically engineered to streamline event exploration and engagement through the effective utilization of user preferences. With a comprehensive range of functionalities, users can seamlessly navigate and participate in diverse events, while event planners gain the ability to create and manage their own events. At its core, the platform employs cosine similarity recommendation algorithms to analyze user preferences, providing personalized event suggestions tailored to individual likes and interests. This approach significantly heightens the probability of user attendance and interaction. In tandem with the recommendation feature, the platform seamlessly integrates Google Maps API, offering a map functionality that enables users to visualize event locations alongside pertinent details. This feature empowers users to efficiently plan their attendance by considering factors such as proximity and accessibility. The overarching goal is to cultivate a vibrant, interactive community by connecting event enthusiasts with organizers. The platform's reliance on Firebase Cloud Function APIs for CRUD operations, such as post creation, reading, updating, and deletion, ensures robust support. Additionally, the recommendation function, based on cosine similarity, is deployed as an API to Firebase Cloud Function. These APIs empower administrators to perform CRUD operations on events, while users can access events based on location and receive recommendations rooted in their favorite events. By seamlessly integrating event recommendations and location-based information, the platform elevates the overall event discovery and participation experience, rendering it more engaging and personalized for users.

Keywords: social events networking application, react-native, cosine similarity recommendation algorithm, Google Maps API, real-time
Introduction
Social events networking platforms have revolutionized the way individuals connect, collaborate, and engage in professional and social settings. In today's society 5.0, the use of social media for networking purposes has become increasingly prevalent, offering unique opportunities for individuals to expand their networks and enhance their careers for governance to agriculture operation (Bhagat et al., 2022).

The Social Events Networking Platform is a specialized application addressing the requirements of event organizers and their members. This application provides an efficient and user-friendly platform for organizers to effortlessly create and publish events, simplifying the process for members and other users to discover and participate in exciting club events (Chee, C., 2022).

The app streamlines the event planning process, enabling planners to manage their events effortlessly, from creation to promotion and oversight. Users can conveniently access all event details, ensuring they remain well-informed and organized. Individual event details, encompassing crucial information such as date, time, venue, and organizer details, are easily accessible. The inclusion of a map feature proves particularly beneficial for users seeking specific venues, guaranteeing a hassle-free navigation experience. This facilitates confident decision-making regarding event attendance based on interests and availability.

Adding a personal touch, the app incorporates a recommendation system that suggests events based on user preferences and past interactions. The inclusion of a favorites feature allows users to curate their event lists and receive tailored recommendations, enhancing the enjoyment and relevance of the discovery process. This personalized approach fosters increased user engagement, ensuring individuals are consistently exposed to events aligning with their interests.

Organizing events for organizations presents notable challenges, characterized by a time-consuming and intricate process. Despite these challenges, there is a noticeable absence of dedicated social media platforms tailored specifically to meet the unique needs of event organizing (Krueger, 2022). The existing solutions often necessitate the use of multiple platforms, resulting in inefficiencies and complexities in event management.

The absence of a dedicated platform focused on event organizing highlights a critical gap that needs attention. Current tools fall short in providing an integrated and streamlined solution, prompting the necessity for a specialized social media app dedicated to event organization. Such an app could offer organizers the ability to efficiently manage events, extend event reach to a broader audience, and foster social interactions among event attendees.

The evident frustration with existing tools emphasizes the clear need for an effective and user-friendly solution in facilitating event planning and organization. The challenges associated with current methods underscore the imperative for a dedicated platform that addresses the unique demands of event organizers and participants.

In addition, the swift expansion of social media platforms has raised concerns about potential adverse effects on face-to-face interactions, psychological well-being, and relationship quality. Understanding and addressing these challenges are essential for effectively harnessing the benefits of social events networking platforms.

The identified problems include the absence of a dedicated social media platform for event organizing, inefficiencies and complexities in current solutions, and the potential negative impacts of rapid social media growth. Recognizing and addressing these challenges are pivotal for the successful development and
implementation of a dedicated Social Events Networking Platform, which holds great promise as an invaluable tool for individuals and clubs passionate about organizing and participating in events. The objective of the application for the social platform for club events is to furnish a dedicated and efficient tool for event organization, aiming to streamline the event planning process and enhance social interaction among planners and event attendees.

**Review of Literature**

Research indicates that social media platforms play a significant role in connecting professionals across various industries. The Theory of Planned Behavior has been used to assess factors influencing event fans’ decisions on social media platforms, highlighting the importance of these platforms as marketing tools for events. Additionally, studies have shown that social media offers unique opportunities for individuals to build their personal brand, establish credibility, and attract potential clients or collaborators. The landscape of event planning and promotion has undergone significant transformations in recent years, propelled by the escalating use of social media, mobile apps, virtual and hybrid events, and AI/ML technologies (Badami et al., 2018). This comprehensive review aims to explore and analyze these trends, highlighting their impact on event management, attendee experiences, and audience reach. The integration of these technologies has ushered in a new era of more streamlined, efficient, and engaging event planning.

Social media platforms have become integral for event organizers, with over 80% utilizing them for marketing and engagement purposes (Doshan & Rupesh, 2022). This underscores the crucial role of social media in enhancing event visibility and attracting participants. The rise of smartphones and mobile internet access has further catalyzed the evolution of event networking, with approximately 86% of event attendees using dedicated apps for real-time updates, information access, and engagement with fellow participants (Event Management Software Industry Report, 2020).

The advent of virtual and hybrid events, particularly following the global pandemic, has been marked by a staggering 1000% growth rate in the virtual events market in 2020 (Badami et al., 2018). This surge in popularity has transcended geographical boundaries, allowing organizers to reach a broader audience and ensuring continuous engagement. However, challenges persist in the form of managing events across multiple platforms, as indicated by 70% of event planners finding it challenging and time-consuming (Steinert-Threlkeld, 2019). This underscores the need for a comprehensive and integrated solution for streamlined event management.

The incorporation of AI and ML technologies into event networking platforms has shown promising results, with personalized event recommendations leading to a 45% increase in attendance rates (Ahmed et al., 2014). Intelligent matchmaking algorithms have also proven effective, with 78% of event attendees reporting successful connections through such features (Steinert-Threlkeld, 2019). Leveraging these capabilities is crucial for optimizing user experiences and fostering meaningful interactions.

Social media influencers have emerged as influential collaborators for event promotion, with 89% of marketers affirming a positive return on investment for event promotions through influencer marketing. This strategy taps into targeted and engaged audiences, creating buzz and expanding event reach. Sustainability practices have gained prominence, with 82% of event attendees considering environmental factors when choosing events (Kawamoto et al., 2013). Event organizers now incorporate green practices to align with societal values.

The integration of AR and VR technologies has revolutionized the event experience, offering
immersive and interactive elements even in virtual settings. Studies indicate a 30% increase in attendee satisfaction and engagement for events incorporating AR/VR technologies. Additionally, chatbots in event management, blockchain-based ticketing systems, live streaming, on-demand content, and gamification elements contribute to enriching and diverse event experiences (Liu et al., 2012; Kayastha et al., 2011; Ball, 2013).

Personalization has become paramount, with 91% of event planners recognizing its impact on attendee satisfaction and event success. The real-time nature of social media has transformed attendee interactions, with over 70% using platforms to share experiences and post live updates during events. Virtual reality enhances physical exhibitions in virtual realms, and offering closed captions and multilingual translations has become essential for global events, ensuring inclusivity (Future Watch, 2013; Ng et al., 2006; & McCreadie et al., 2013).

However, data security and privacy concerns have surfaced, with 67% of event attendees expressing worries about personal information during virtual events. Implementing robust data protection measures is imperative to build trust and encourage higher event participation rates (Wenyue, et al., 2019; Hui et al., 2019).

Incorporating these trends and strategies into event planning and promotion can enhance engagement, inclusivity, and overall success. Technology continues to shape the future of events, with ongoing innovations anticipated in event planning and execution. In conclusion, the dynamic evolution of event planning, driven by technological advancements, showcases the potential for continued improvements and innovations in the field (Hui et al., 2019; Jin-Hui et al., 2019).

**Methodology**

The subsequent step in this process is system design, which entails elucidating data requirements, outlining the system architecture, and ensuring a user-friendly interface that aptly caters to these pivotal functionalities (Mishara, 2023).

**Requirement Analysis**

**i. Functional Requirements**

- **User Authentication**: Both planners and general users must have the capability to register and log in.
- **Account Management**: Planners and general users should be equipped to manage their respective accounts seamlessly.
- **Event Creation and Posting**: Planners must have the ability to create and post events effortlessly.
- **User Participation**: General users should express interest and participation in events by liking event posts.
- **Event Viewing**: General users must be able to view recommended event posts and their associated locations.

**ii. Non-functional Requirements**

- **Usability**: The system should boast an intuitive and user-friendly interface, ensuring ease of use and navigation.
- **Performance**: The system must be swift and responsive to provide an efficient user experience.
- **Reliability**: The system should exhibit high reliability, ensuring consistent and dependable performance.
- **Security**: Stringent security measures should be in place to safeguard user data and system integrity.
- **Data Quality**: The system should maintain high-quality data to enhance the overall user experience.

This systematic breakdown of requirements, both functional and non-functional, serves as a foundational step for the subsequent stages of system design and development. It sets the stage
for creating an event recommendation app that is not only feature-rich but also ensures a seamless and secure experience for both planners and general users (Mishra et al., 2023; Pokhrel et al., 2021; & Jha et al., 2023).

**Figure 1**

*Object Modelling: Object and Class Diagram*

<table>
<thead>
<tr>
<th>Admin</th>
<th>Event</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ admin_ID</td>
<td>+ admin_name</td>
<td>+ user_id</td>
</tr>
<tr>
<td>+ admin_email</td>
<td>+ admin_password</td>
<td>+ user_name</td>
</tr>
<tr>
<td>+ createEvent()</td>
<td>+ collegeName</td>
<td>+ user_email</td>
</tr>
<tr>
<td>+ updateEvent()</td>
<td>+ date</td>
<td>+ description</td>
</tr>
<tr>
<td>+ deleteEvent()</td>
<td>+ genre</td>
<td>+ viewDetails()</td>
</tr>
<tr>
<td>+ signIn()</td>
<td>+ image</td>
<td>+ viewRecommendation()</td>
</tr>
<tr>
<td>+ signOut()</td>
<td>+ location</td>
<td>+ add_to_favorites()</td>
</tr>
</tbody>
</table>

The interconnections among the various classes in the system are elucidated as follows from Figure 1:

1. **Admin Class**
   **Responsibilities:** Create, update, and delete Events.

2. **Event Class**
   - **Responsibilities:** Hold details of events.
   - **Relationships:** Many-to-Many with Admin: An Admin can create multiple Events, and an Event can be created by multiple Admins.
   - **Many-to-Many with User:** A User can view multiple Events, and an Event can be viewed by multiple Users.

3. **User Class**
   **Responsibilities:** View event details, View event recommendations, Add and remove Events from their favorites list, and Sign in and sign out of the system.

4. **Relationships**
   **Many-to-Many with Event**
   - A User can view multiple Events, and an Event can be viewed by multiple Users.

   - This comprehensive description highlights the key responsibilities and relationships between the classes in the system:
     - **Admins:** Primarily responsible for the creation, update, and deletion of Events. They participate in a many-to-many relationship with Events, signifying that an Admin can create multiple Events, and conversely, an Event can be created by multiple Admins.
     - **Events:** Central to holding event details, Events participate in many-to-many relationships with both Admins and Users. This denotes that an Admin can create multiple Events, and an Event can be viewed by multiple Users.
     - **Users:** Engaged in various activities such as viewing event details, recommendations, and managing their favorite events. Users also sign in and sign out of the system. The many-to-many relationship with Events signifies that a User can view multiple Events, and an Event can be viewed by multiple Users.
This structured representation facilitates a comprehensive understanding of the roles and connections between the classes, laying the foundation for effective system design and implementation.

**Dynamic Modelling: State and Sequence Diagram**

The platform encompasses a comprehensive set of states as shown in figure 2 and transitions to capture the user journey and interactions:

**Figure 2**  
*State Diagram*

![State Diagram](image)

**States**

1. **Unregistered**
   **Description:** The initial state where the user is not registered on the platform.

2. **Registered**
   **Description:** The user has successfully signed up for an account.

3. **Planner**
   **Description:** The user has logged in and chosen to plan an event.

4. **No Event Created:**
   **Description:** The default state within the Planner state when the user has not yet created an event.

5. **Event Created**
   **Description:** The user has successfully created an event, transitioning from the No Event Created state.

6. **Post Created**
   **Description:** The user has crafted and posted details about the event, transitioning from the Event Created state.

7. **Transitions**
   **User signs up**
   **Description:** Transitions the user from the Unregistered state to the Registered state.

8. **User logs in:**
   **Description:** Moves the user from the Registered state to the Planner state.
9. **No Event Created**  
**Description:** Default transition from the Planner state, leading back to the Planner state.

10. **Provide Event Details**  
**Description:** Transitions the user from the Planner state to the Event Created state, triggered when the user provides event details.

11. **Create Event**  
**Description:** Transitions the user from the Event Created state to the Post Created state, activated when the user creates the event.

12. **Edit Post**  
**Description:** Sends the user from the Post Created state back to the Event Created state, activated when the user edits the post about the event.

13. **Delete Post**  
**Description:** Shifts the user from the Post Created state back to the Planner state, triggered when the user deletes the post about the event.

14. **View Post**  
**Description:** Transitions the user from the Post Created state to a state where they can view the post about the event (not explicitly shown in the diagram but implied).

15. **Notify users about the changes made**  
**Description:** Occurs after the user edits or deletes the post about the event (not explicitly shown in the diagram but implied).

This detailed set of states and transitions provides a comprehensive framework for understanding the various stages and user actions within the platform, facilitating effective design and implementation (Mishra et al., 2023; Pokhrel et al., 2021; & Jha et al, 2023).

**Figure 3**  
Sequence Diagram

The application sequence unfolds as follows:

- **User opens the app:** The sequence initiates with the user opening the React Native app (ReactNativeApp).
- **Sign-in request:** ReactNativeApp triggers a sign-in request directed towards the authentication component (FirebaseAuth). The activation bar for FirebaseAuth signifies active processing of the sign-in request.
• **Successful sign-in:** FirebaseAuth processes the sign-in request, and upon successful authentication, transmits a response back to ReactNativeApp. Deactivation of FirebaseAuth signals the completion of the sign-in process.

• **Request nearby events:** Authenticated ReactNativeApp sends a request for nearby events to the EventsListing component. The activation bar for EventsListing indicates active processing of the request.

• **List of nearby events:** EventsListing processes the request and forwards a list of nearby events to ReactNativeApp. Deactivation of EventsListing signals the completion of fetching nearby events.

• **User views nearby events:** The user interacts with ReactNativeApp to view the list of nearby events.

• **Get favorite events:** ReactNativeApp requests the list of favorite events from the Favorites component. The activation bar for Favorites indicates active processing of the request.

• **List of favorite events:** Favorites processes the request and transmits the list of favorite events back to ReactNativeApp. Deactivation of Favorites marks the completion of fetching favorite events.

• **Get recommended events:** ReactNativeApp requests recommended events from the Recommendations component based on the user's favorites. The activation bar for Recommendations indicates active processing of the request.

• **List of recommended events:** Recommendations processes the request and sends back a list of recommended events to ReactNativeApp. Deactivation of Recommendations indicates the completion of fetching recommended events.

• **End of sequence:** The sequence concludes, and all components are deactivated.

**Figure 4**
*Component Diagram (Mishra & Jha 2023; Pokhrel et al., 2021)*

**Figure 5**
*Deployment Diagram*
**Algorithms Details**

Cosine Algorithm for Event Recommendations

The recommendation screen employs an advanced cosine algorithm to intelligently suggest events tailored to the user's preferences. This algorithm assesses the cosine similarity between the chosen event and others in the database, taking into account critical factors such as genre and ratings. Cosine similarity, ranging from -1 to 1, quantifies the likeness between two vectors, providing a robust measure of their alignment. The formula for Cosine Similarity Algorithm is given by:

$$\text{sim}(\vec{x}, \vec{y}) = \cos \theta = \frac{\vec{x} \cdot \vec{y}}{||\vec{x}|| \times ||\vec{y}||}$$

By analyzing the cosine similarities of different events with the selected one, the algorithm discerns events closely aligned with the user's interests. The cosine algorithm calculates the cosine of the angle between two vectors, representing the event's genre and ratings. This measure gauges the similarity of their orientations, irrespective of vector magnitude. A higher cosine similarity signifies a stronger alignment between two events.

The application's recommendation screen then showcases a curated list of events closely matching the user's preferences, leveraging the power of the cosine algorithm. This strategic use enhances the user experience by offering events likely to pique the user's interest, fostering seamless exploration and discovery within the social events networking platform.

**Pseudocode for Cosine Similarity:**

```javascript
function cosineSimilarity(item1, item2) { if (item1.length !== item2.length) { throw new Error("Item vectors must have the same length"); } let dotProduct = 0; let magnitude1 = 0; let magnitude2 = 0; for (let i = 0; i < item1.length; i++) { dotProduct += item1[i] * item2[i]; magnitude1 += item1[i] * item1[i]; magnitude2 += item2[i] * item2[i]; } magnitude1 = Math.sqrt(magnitude1); magnitude2 = Math.sqrt(magnitude2); if (magnitude1 === 0 || magnitude2 === 0) { return 0; } return dotProduct / (magnitude1 * magnitude2); }
```

**Results and Discussion**

The testing phase for the "Social Events Networking Platform" plays a pivotal role in the software development life cycle, systematically examining and validating multiple facets of the application to ensure its robust functionality, usability, performance, and security.

With a primary focus on validating the correctness, reliability, and security of the platform, the testing process is meticulously designed to unearth and rectify any discrepancies or issues, ensuring adherence to specified requirements. The overarching objectives encompass:

Functionality Verification: Rigorous testing is conducted to confirm that all features, spanning event creation, editing, deletion, and user interactions, operate seamlessly according to their intended functionality as shown in table 1.
### Table 1
**API Endpoint Function Test Cases**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Function</th>
<th>Input</th>
<th>Expected Output</th>
<th>Result</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>createEvent</td>
<td>Valid event data in the request body</td>
<td>HTTP 201, JSON with event ID, and success message</td>
<td>HTTP 201, `{ &quot;id&quot;: &quot;xyz123&quot;, &quot;message&quot;: &quot;Event created successfully&quot; }</td>
<td>Passed</td>
</tr>
<tr>
<td>2</td>
<td>createEvent</td>
<td>Missing event data in the request body</td>
<td>HTTP 500 and error message indicating failure to create event</td>
<td>HTTP 500, `{ &quot;error&quot;: &quot;Failed to create event&quot; }</td>
<td>Passed</td>
</tr>
<tr>
<td>3</td>
<td>getEvents</td>
<td>Valid location parameter in the query</td>
<td>HTTP 200, JSON array of events in specified location</td>
<td>HTTP 200, <code>{ &quot;id&quot;: &quot;event1&quot;, &quot;name&quot;: &quot;Event A&quot; }, </code>{ &quot;id&quot;: &quot;event2&quot;, &quot;name&quot;: &quot;Event B&quot; } }</td>
<td>Passed</td>
</tr>
<tr>
<td>4</td>
<td>getEvents</td>
<td>Invalid location parameter in the query</td>
<td>HTTP 500 and error message indicating failure to fetch events</td>
<td>HTTP 500, `{ &quot;error&quot;: &quot;Failed to fetch events&quot; }</td>
<td>Passed</td>
</tr>
<tr>
<td>5</td>
<td>getEvent</td>
<td>Valid event ID in the query</td>
<td>HTTP 200 and JSON object with event details</td>
<td>HTTP 200, `{ &quot;id&quot;: &quot;event1&quot;, &quot;name&quot;: &quot;Event A&quot; } }</td>
<td>Passed</td>
</tr>
<tr>
<td>6</td>
<td>getEvent</td>
<td>Invalid event ID in the query</td>
<td>HTTP 404 and error message indicating event not found</td>
<td>HTTP 404, `{ &quot;error&quot;: &quot;Event not found&quot; }</td>
<td>Passed</td>
</tr>
<tr>
<td>7</td>
<td>updateEvent</td>
<td>Valid event ID and updated data</td>
<td>HTTP 200 and success message</td>
<td>HTTP 200, `{ &quot;message&quot;: &quot;Event updated successfully&quot; }</td>
<td>Passed</td>
</tr>
<tr>
<td>8</td>
<td>updateEvent</td>
<td>Invalid event ID in the query</td>
<td>HTTP 500 and error message indicating failure to update event</td>
<td>HTTP 500, `{ &quot;error&quot;: &quot;Failed to update event&quot; }</td>
<td>Passed</td>
</tr>
<tr>
<td>9</td>
<td>deleteEvent</td>
<td>Valid event ID in the query</td>
<td>HTTP 200 and success message</td>
<td>HTTP 200, `{ &quot;message&quot;: &quot;Event deleted successfully&quot; }</td>
<td>Passed</td>
</tr>
<tr>
<td>10</td>
<td>deleteEvent</td>
<td>Invalid event ID in the query</td>
<td>HTTP 500 and error message indicating failure to delete event</td>
<td>HTTP 500, `{ &quot;error&quot;: &quot;Failed to delete event&quot; }</td>
<td>Passed</td>
</tr>
<tr>
<td>11</td>
<td>get Recommendations</td>
<td>Valid event IDs in the query</td>
<td>HTTP 200 and JSON array of recommended events</td>
<td>HTTP 200, `{ { &quot;selectedEvent&quot;: { &quot;id&quot;: &quot;event1&quot;, &quot;name&quot;: &quot;Event A&quot; }, &quot;recommendations&quot;: [ { &quot;id&quot;: &quot;event2&quot;, &quot;name&quot;: &quot;Event B&quot; }, { &quot;id&quot;: &quot;event3&quot;, &quot;name&quot;: &quot;Event C&quot; } ] } }</td>
<td>Passed</td>
</tr>
<tr>
<td>12</td>
<td>get Recommendations</td>
<td>Missing event IDs in the query</td>
<td>HTTP 400 and error message indicating missing event IDs</td>
<td>HTTP 400, `{ &quot;error&quot;: &quot;Missing event IDs in the query parameters.&quot; }</td>
<td>Passed</td>
</tr>
</tbody>
</table>
Usability Assessment
The user interface design and overall user experience undergo an evaluation to ascertain that the application is intuitively navigable for both administrators and regular users, fostering a user-friendly interaction.

Performance Evaluation
The responsiveness and efficiency of the application are thoroughly assessed, with a specific focus on key functionalities such as fetching nearby events, loading event details, and generating event recommendations. This evaluation ensures optimal performance under varying conditions.

Security Validation
A critical aspect involves identifying and addressing potential vulnerabilities in data handling, user authentication, and authorization mechanisms. This meticulous validation process aims to fortify the platform against security threats, safeguarding both user and event data.

By fulfilling these comprehensive objectives, the testing phase contributes significantly to the overall quality assurance of the "Social Events Networking Platform," guaranteeing a reliable, user-friendly, and secure environment for event organizers and participants as result shown in table 2.

Table 2
Login Test Cases

<table>
<thead>
<tr>
<th>SN</th>
<th>Test Case</th>
<th>Input</th>
<th>Expected Output</th>
<th>Result</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Valid Credentials Login</td>
<td>Valid email and password</td>
<td>User successfully logged in</td>
<td>Navigate to HomeScreen</td>
<td>Passed</td>
</tr>
<tr>
<td>2</td>
<td>Invalid Email</td>
<td>Invalid email format (e.g., &quot;invalidemail&quot;)</td>
<td>Validation error for email displayed</td>
<td>Redirect to LoginScreen</td>
<td>Passed</td>
</tr>
<tr>
<td>3</td>
<td>Empty Email</td>
<td>Empty email field</td>
<td>Validation error for email displayed</td>
<td>Redirect to LoginScreen</td>
<td>Passed</td>
</tr>
<tr>
<td>4</td>
<td>Invalid Password</td>
<td>Invalid password (e.g., too short or weak)</td>
<td>Validation error for password displayed</td>
<td>Redirect to LoginScreen</td>
<td>Passed</td>
</tr>
<tr>
<td>5</td>
<td>Empty Password</td>
<td>Empty password field</td>
<td>Validation error for password displayed</td>
<td>Redirect to LoginScreen</td>
<td>Passed</td>
</tr>
<tr>
<td>6</td>
<td>Incorrect Credentials</td>
<td>Correct email but incorrect password</td>
<td>Error message indicating incorrect credentials</td>
<td>Redirect to LoginScreen</td>
<td>Passed</td>
</tr>
<tr>
<td>7</td>
<td>ValidEmail (Special Char)</td>
<td>Valid email with special characters</td>
<td>User successfully logged in</td>
<td>Redirect to LoginScreen</td>
<td>Passed</td>
</tr>
<tr>
<td>8</td>
<td>Forgot Password Link</td>
<td>Click on &quot;Forgot Password&quot; link</td>
<td>Navigate to Forgot Password screen</td>
<td>Navigate to ForgotScreen</td>
<td>Passed</td>
</tr>
<tr>
<td>9</td>
<td>Create Account</td>
<td>Click on &quot;Signup&quot;</td>
<td>Navigate to Signup screen</td>
<td>Navigate to SignupScreen</td>
<td>Passed</td>
</tr>
<tr>
<td>10</td>
<td>Login Button Press</td>
<td>Click Login button without entering credentials</td>
<td>Validation errors for both email and password</td>
<td>Redirect to LoginScreen</td>
<td>Passed</td>
</tr>
</tbody>
</table>

Test cases for System Testing
System testing is a comprehensive evaluation of the entire software system to verify that all components work together seamlessly, meeting specified requirements. It aims to assess the system's functionality, performance, and reliability in a real-world environment before deployment as shown in table 3.
<table>
<thead>
<tr>
<th>SN</th>
<th>Test Case</th>
<th>Input</th>
<th>Expected Output</th>
<th>Result</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>User Login</td>
<td>Valid email and password</td>
<td>User logged in successfully, directed to event list</td>
<td>Navigate to HomeScreen</td>
<td>Passed</td>
</tr>
<tr>
<td>2.</td>
<td>Invalid Login Attempt</td>
<td>Invalid email or password</td>
<td>Error message indicating incorrect credentials</td>
<td>Redirect to LoginScreen</td>
<td>Passed</td>
</tr>
<tr>
<td>3.</td>
<td>User Signup</td>
<td>Valid email, password, and details</td>
<td>New user account created, directed to event list</td>
<td>Navigate to SignupScreen</td>
<td>Passed</td>
</tr>
<tr>
<td>4.</td>
<td>Forgot Password</td>
<td>User enters registered email, requests password reset</td>
<td>Password reset email received, guided through reset process</td>
<td>Navigate to ForgotPassword Screen</td>
<td>Passed</td>
</tr>
<tr>
<td>5.</td>
<td>View Events</td>
<td>User logs in successfully</td>
<td>List of events displayed</td>
<td>List of Events displayed on Cards</td>
<td>Passed</td>
</tr>
<tr>
<td>6.</td>
<td>Event Details</td>
<td>User clicks on a specific event</td>
<td>Detailed event information shown</td>
<td>Shown Details about the Event</td>
<td>Passed</td>
</tr>
<tr>
<td>7.</td>
<td>Search Events by Location</td>
<td>User enters location in the search bar</td>
<td>Event list filtered based on entered location</td>
<td>Filtered Existing Cards based on Location</td>
<td>Passed</td>
</tr>
<tr>
<td>8.</td>
<td>Favorite an Event</td>
<td>User marks an event as a favorite</td>
<td>Event added to user's favorites list</td>
<td>Favourite Button got Highlighted</td>
<td>Passed</td>
</tr>
<tr>
<td>9.</td>
<td>Generate Recommendations</td>
<td>User has favored events</td>
<td>Recommendations generated based on favorites</td>
<td>New List of Recommended Events on Recommendation Tab</td>
<td>Passed</td>
</tr>
<tr>
<td>10.</td>
<td>Admin Login</td>
<td>Admin credentials</td>
<td>Admin logged in, directed to admin dashboard</td>
<td>Navigate to AdminHomeScreen</td>
<td>Passed</td>
</tr>
<tr>
<td>11.</td>
<td>Create Event</td>
<td>Admin creates a new event using API</td>
<td>New event added to the list of events</td>
<td>New Event added to firestore and shown on the Events List</td>
<td>Passed</td>
</tr>
<tr>
<td>12.</td>
<td>Edit Event</td>
<td>Admin edits an existing event using API</td>
<td>Changes reflected in event details</td>
<td>Event got Edited on firestore and Events List</td>
<td>Passed</td>
</tr>
<tr>
<td>13.</td>
<td>Delete Event</td>
<td>Admin deletes an event using API</td>
<td>Event removed from the list</td>
<td>Event got deleted from firestore and disappeared from the Events List</td>
<td>Passed</td>
</tr>
</tbody>
</table>

Embarking on the journey of developing a Social Event Networking Platform has been a rich and enlightening experience, yielding valuable insights essential for the platform's success. One of the pivotal lessons learned was the centrality of user-centric design, emphasizing the creation of user-friendly and intuitive interfaces.
of an interface that not only proves intuitive but also facilitates effortless event discovery and connection-building. The commitment to simplicity and user-friendliness emerged as a guiding principle, advocating for a minimalist design approach to heighten accessibility.

The backbone of efficient backend systems proved crucial, recognizing the dynamic nature of event networking. This underscored the importance of scalable infrastructure and responsive algorithms in ensuring a seamless user experience. Despite the absence of formal feedback mechanisms, the emphasis on continuous user engagement emerged as a key driver, leading to the incorporation of features aimed at sustaining interaction and participation.

The paramount significance of data security and privacy, particularly concerning user connections, became non-negotiable. This necessitated stringent adherence to regulations and the implementation of robust protective measures to instill user trust. Adopting an iterative development approach facilitated the introduction of new features and enhancements, ensuring the platform's dynamism and alignment with evolving user needs.

The absence of notifications and formal feedback mechanisms highlighted the critical need to anticipate user expectations and proactively address potential pain points. Throughout the project, these lessons have collectively shaped a Social Event Networking Platform that not only fulfills its objectives but also provides a meaningful, secure, and dynamic environment for users to connect and engage.

The findings of this research paper suggest that social events networking platforms offer numerous benefits for professionals, including expanded networking opportunities, increased visibility, and access to a global audience. However, challenges such as information overload, privacy concerns, and online harassment also exist, highlighting the need for individuals to navigate these platforms mindfully.

Conclusion

The Social Events Networking Platform developed for this project represents a holistic solution for seamless event discovery and active participation. The platform's utilization of user preferences and likes enables the delivery of personalized event recommendations, significantly increasing the likelihood of user attendance and engagement. The inclusion of a map feature further elevates the user experience by providing easy access to event locations and relevant details, thereby facilitating efficient event planning and enhancing overall accessibility.

The intuitiveness of the platform's interface contributes to a user-friendly environment, catering to both event participants and organizers. This design ensures effortless navigation through events, facilitating effective event creation and management. In essence, the Social Events Networking Platform endeavors to cultivate a vibrant and closely-knit community by uniting event enthusiasts and organizers. It streamlines the event discovery process, champions personalized recommendations, and elevates the overall event experience for all stakeholders involved.

In a broader context, social events networking platforms have undergone a transformative impact on how individuals connect, collaborate, and establish relationships. To harness the full potential of these platforms, individuals are encouraged to prioritize authenticity, engage meaningfully with their audience, and strike a balance between online and offline interactions. Key recommendations include setting clear boundaries for personal social media use, staying abreast of platform policies, and consistently evaluating the impact of social media on personal well-being.

Acknowledgement

The authors are grateful to Dr. Anjay Kumar Mishra and Mohan Saud for their continuous motivation and support.
References


