

AllEvently: Requirements Document

By: Benjamin Bruyns, Peggy Lewis, and Spenser Morey

Table of Contents

Table of Contents.....	2
Project Name	3
Team Member Names	3
Abstract	3
Tools & Technologies	3
Requirements List.....	3
1. Login Page	3
2. Sign-Up Page	4
3. Password Reset Page	4
4. Update Password Page.....	5
5. User Events Page	5
8. Event Creation Page	6
9. Event Chat	8
10. Guest List Page	8
11. Invitation Page	8
12. Print Invitation Page.....	10
13. Navigation Menu	10
14. Event	10
15. Design Requirements	11
16. Database Requirements	11
17. Optional Requirements.....	11
Updated Timeline.....	11
Appendix.....	12
Figure 1: Team Member Names	12
Figure 2: AllEvently Tech-Stack	12
Figure 3: Tentative Schedule*	13

Project Name

Our event-planning web application's primary goal is to manage the coordination of event planning. The core concepts of "all events" and "event management" stood out during brainstorming. By merging these concepts, we arrived at the name **AllEvently** for our application. This name effectively conveys our platform's purpose of centralizing and managing all aspects of event planning.

Team Member Names

See Figure 1 in the appendix for a detailed description.

Abstract

When events are held, it can take time to determine the exact number of people attending them. So, invitations are sent out to invite guests and to determine how many people will be in attendance. However, if information is required in the RSVP, compiling information from the invitees takes much work.

To solve this problem, we are creating AllEvently. This program will be used to create events requiring attendees' information. The program will then be able to receive RSVPs from invitees. The data will then be stored, and the event host can compile information from the RSVPs.

Tools & Technologies

See Figure 2 in the appendix for a detailed description.

Requirements List

1. Login Page

- 1.1. A logo
- 1.2. An email address label
- 1.3. An email address field
 - 1.3.1. Email must include an @ symbol
 - 1.3.2. Email must include a period
- 1.4. A password label
- 1.5. A password field
- 1.6. A password reset hyperlink
 - 1.6.1. Directs the user to Password Reset Page
- 1.7. A create account hyperlink
 - 1.7.1. Directs the user to Sign-Up page

1.8. A Google login button

1.8.1. Upon success, directs the user to the Event Page

1.9. A login button

1.9.1. Upon clicking, validate credentials from email and password fields

1.9.1.1. On validation, direct the user to the Event Page

1.9.1.2. On failure, show incorrect login dialogue box

1.9.1.2.1. A logo

1.9.1.2.2. A 'try again' button

1.9.1.2.2.1. Dismiss dialogue box when pressed

1.9.1.2.3. A reset password button

1.9.1.2.3.1. Directs the user to Password Reset Page when pressed

2. Sign-Up Page

2.1. A logo

2.2. A first name label

2.3. A field for a first name

2.4. A last name label

2.5. A field for a last name

2.6. An email address label

2.7. A field for email address

2.7.1. Email must include an @ symbol

2.7.2. Email must include a period

2.8. A password label

2.9. A field for the password

2.10. A confirmed password label

2.11. A field for a confirmed password

2.12. A hyperlink to the Login page

2.13. A Sign-Up button

2.13.1. Upon press, validate user credentials

2.13.1.1. Upon validation, direct the user to the Event Page

2.13.1.2. Upon failure, a dialogue box will appear

2.13.1.2.1. A logo

2.13.1.2.2. A message about a failed account creation attempt

2.13.1.2.3. A button to confirm the message and close the dialogue box

2.14. A Google Sign-Up button

2.14.1. Upon success, direct the user to the Event Page

3. Password Reset Page

3.1. A logo

3.2. An email address label

3.3. A field for email

3.4. A reset password button

3.4.1. Sends password reset link to email provided

3.4.1.1. Then show the email sent dialogue box

3.4.1.1.1. A logo

3.4.1.1.2. Email sent text

3.4.1.1.3. An ok button

3.4.1.1.3.1. Directs the user to Login Page

3.5. A cancel button

3.5.1. Directs the user to the Login Page

4. Update Password Page

4.1. A logo

4.2. A new password label.

4.3. A field for the new password

4.4. A confirm password label

4.5. A field for confirming password

4.6. An update password button

4.6.1. Get the email address from HTML and get a request in the link

4.6.2. Assign the new password to the corresponding email

4.6.3. Displays message to the user for a successful update

5. User Events Page

5.1. A logo

5.2. A navigation menu

5.2.1. See requirement 13

5.3. A Create event button

5.3.1. Directs the user to the Event Creation page

5.4. A list of events attended/hosted by the current user or that they have been invited to

5.4.1. See requirement 14

5.5. Search bar

5.5.1. Filters events by name

6. Public Events Page

6.1. A logo

6.2. A navigation menu

6.2.1. See requirement 13

6.3. A Create event button

6.3.1. Directs the user to the Event Creation Page

6.4. A list of all public events

6.4.1. See requirement 14

6.5. Filter option by month

6.6. Filter option by week

6.7. Filter option by day

7. Account Page

- 7.1. A Logo
- 7.2. A Navigation Menu
 - 7.2.1. See requirement 13
- 7.3. A first name label
- 7.4. A field for the first name
- 7.5. A last name label
- 7.6. A field for the last name
- 7.7. A button to save
 - 7.7.1. Upon press, update user account information

8. Event Creation Page

- 8.1. A logo
- 8.2. A label for the event name and note addition
- 8.3. An accordion menu for entering event name and note
 - 8.3.1. Upon clicking, expand fields for event name and note
- 8.4. A label for the date and time
- 8.5. An accordion menu for entering event date and time
 - 8.5.1. Upon clicking, expand fields for event date and time
 - 8.5.1.1. Field for start date and time
 - 8.5.1.1.1. MM/DD/YYYY HH:mm AM/PM format
 - 8.5.1.2. Optional All-Day event selection
 - 8.5.1.3. Optional end date and time selection
 - 8.5.1.3.1. Field for end date and time
 - 8.5.1.3.2. MM/DD/YYYY HH:mm AM/PM format
 - 8.5.1.4. Time Zone selection
 - 8.5.1.5. Recurring event selection
 - 8.5.1.5.1. A label for the “Repeats” section
 - 8.5.1.5.1.1. A label for how often the event will occur (“every...”)
 - 8.5.1.5.1.2. A field for a numerical input for how often the event will occur (1...*)
 - 8.5.1.5.1.3. A field for a timeframe for how often the event will occur (“day,” “week,” “month,” “year”)
 - 8.5.1.5.2. A label for the “Ends” section
 - 8.5.1.5.2.1. Selection options for “never,” “on [date selection],” or “after [numerical selection] times.”
- 8.6. A label for event location
- 8.7. An accordion menu for entering the event location
 - 8.7.1. Upon clicking, expand the field for the event location
 - 8.7.2. Label for address
 - 8.7.3. Field for address
 - 8.7.3.1. Google can verify input from the address field.

- 8.7.3.1.1. If this field is changed after the initial creation of the event, all invited guests or guests in attendance will be notified
 - 8.8. A label for event settings
 - 8.9. An accordion menu for entering event settings
 - 8.9.1. Upon clicking, expand the field for the event settings
 - 8.9.1.1. Public/Private setting
 - 8.9.1.1.1. If the private setting is selected, a list of invited email addresses will be required
 - 8.9.1.1.1.1. Upon event creation, email the guests on the list.
 - 8.9.1.1.1.2. If this list is altered after the initial creation of the event, all people removed from the list will be notified
 - 8.9.1.2. Guest Settings
 - 8.9.1.2.1. A label to request child count from guest
 - 8.9.1.2.2. A field for the host to enable the child count request
 - 8.9.1.2.2.1. If enabled, the guest will be prompted for a numerical value for the number of children during RSVP
 - 8.9.1.2.3. A label to limit additional guests
 - 8.9.1.2.4. A field for the host to enable limiting additional guests
 - 8.9.1.2.4.1. If enabled, the guest will be prompted for a numerical value for the number of additional guests during RSVP
 - 8.9.1.2.5. A label to allow guests to RSVP
 - 8.9.1.2.6. A field for the host to enable the ability to RSVP
 - 8.9.1.2.6.1. If enabled, the guest will have the option to RSVP on the invitation
 - 8.9.1.2.7. A label to allow guests to share
 - 8.9.1.2.8. A field for the host to enable guests to share the event
 - 8.9.1.2.8.1. If enabled, the guest will be able to share the invitation
 - 8.9.1.3. Host Settings
 - 8.9.1.3.1. A label for the host's first and last name
 - 8.9.1.3.2. A field for the host to change the first or last name
 - 8.9.1.3.3. A label for email notifications about the event
 - 8.9.1.3.4. A field for the host to enable the notifications
 - 8.9.1.4. Link Settings
 - 8.9.1.4.1. A list of customizable links the host can edit
 - 8.10. A label for the event theme
 - 8.11. An accordion menu for entering the event theme
 - 8.11.1. Upon clicking, expand the field for the event theme
 - 8.11.1.1. Theme menu
 - 8.11.1.1.1. Image selection menu
 - 8.11.1.1.2. Layout/Customization options
 - 8.11.1.1.2.1. Font for invitation text
 - 8.11.1.1.2.2. Color for information text
 - 8.11.1.1.2.3. Text background color
 - 8.11.1.1.2.4. Opens new tab that shows a pdf of the invitation
- 8.12. A save button

- 8.12.1. Check if the required fields have contents
- 8.13. A preview button
- 8.14. A send button
 - 8.14.1. Opens guest list
 - 8.14.2. See requirement 10

9. Event Chat

- 9.1. A list of every message in the event chat
 - 9.1.1. The message will include the message sender
 - 9.1.2. The message will include message content
 - 9.1.3. The message will include the sent timestamp
- 9.2. Field for new message
- 9.3. Send message button
 - 9.3.1. Adds the message to the message list
 - 9.3.1.1. Message content is the content of the new message field
 - 9.3.1.2. The message sender is the current user
 - 9.3.1.3. The new message field is emptied

10. Guest List Page

- 10.1. A logo
- 10.2. List of all guests
 - 10.2.1. Guests will be listed by name
 - 10.2.1.1. The number of people attending with a guest will be included
 - 10.2.2. Guests can be removed from the list
- 10.3. A guest search bar text field
 - 10.3.1. Filters guests by search bar input
- 10.4. Label for total guests attending
- 10.5. Label for total RSVPs pending
- 10.6. Label for total invitation rejections

11. Invitation Page

- 11.1. An image for this event
- 11.2. A label for the Event name
- 11.3. A label for the Host's name
- 11.4. A label for the Host's email address
- 11.5. A date for the event
- 11.6. A start time of the event
- 11.7. If enabled, an end time of the event
- 11.8. If applicable, recurring event information
- 11.9. A label to display the current RSVP, "Attending," "Not Attending," or "Haven't Responded Yet."
- 11.10. A Google Mapped address for the event
- 11.11. Any additional links added by the host during event creation
- 11.12. A note if the host added one during the event creation

11.13. A link to the Chat on the webpage

11.13.1. The user must be logged in or create an account to access the chat for an event

11.14. If enabled during event creation, a label for sharing the event

11.15. If enabled during event creation, a field for sharing the event

11.16. An accept button

11.16.1. If the current user is logged in:

11.16.1.1. If enabled by the host, prompt the guest for a count of additional guests

11.16.1.2. If enabled by the host, prompt the guest for a count of children

11.16.1.3. A field for an optional note

11.16.1.3.1. Note is only accessible by the host in the event chat

11.16.1.3.2. Note appears green and displays “Accepted” if guest accepts the invite

11.16.1.4. Adds the current user to the accepted group on the guest list for this

11.16.1.5. Directs the user to the User Events Page

11.16.2. Adds the current user to the accepted group on the guest list for this event

11.17. A decline button

11.17.1. If the current user is logged in:

11.17.1.1. A field for an optional note

11.17.1.1.1. Note is only accessible by the host in the event chat

11.17.1.1.2. Note appears red and displays “Regrets” if the guest declined the invite

11.17.1.2. Adds the current user to the declined group in the guest list for this event

11.18. If the current user is not logged in

11.18.1. If enabled by the host, prompt the guest for a count of additional guests

11.18.2. If enabled by the host, prompt the guest for a count of children

11.18.3. A guest's first name label

11.18.4. A guest's first name field

11.18.5. A guest's last name label

11.18.6. A guest's last name field

11.18.7. An email address label

11.18.8. There will be an email address field

11.18.8.1. The field must contain a “@” symbol

11.18.8.2. The field must contain a period

11.18.9. A field for an optional note

11.18.9.1. Note is only accessible by the host in the event chat

11.18.9.2. Note appears green and displays “Accepted” if guest accepts the invite

11.18.9.3. Note appears red and displays “Regrets” if the guest declined the invite

11.18.10. These fields will define the current user information

11.19. Else, the current user is defined by user information

11.20. A login button

11.20.1. Directs the user to the Login page

11.20.2. After login, the event should be included in the user's list of events

12. Print Invitation Page

12.1. A visual copy of the invitation

12.2. A download pdf button

12.2.1. Downloads pdf of the invitation with the QR code on it

12.2.1.1. The QR code directs the user to the Invitation page for this event

13. Navigation Menu

13.1. A logo

13.2. A hyperlink label containing the current user's first and last names

13.2.1. Directs the user to the Account Page

13.3. An event button

13.3.1. Directs the user to the User Event Page

13.4. A public events button

13.4.1. Directs the user to the Public Events Page

13.5. A sign-out button

13.5.1. Directs the user to the Login Page

14. Event

14.1. Label for event name

14.2. Label for public/private

14.3. Link for event location

14.3.1. Directs the user to address in Google Maps

14.4. Label for event date

14.5. Label for the event host's name

14.6. The current user can host events

14.6.1. A cancel event button

14.6.1.1. If the event is canceled, all guests in attendance will be notified via email

14.6.1.2. If the event is private, all guests on the invite list will be notified

14.6.2. Contains the number of attending people

14.6.3. A print invitation button

14.6.3.1. Directs the user to Print Invitation Page for this event

14.6.4. A guest list button

14.6.4.1. Directs the user to the Guest List Page for this event

14.6.5. A button for the event chat

14.6.5.1. Directs the user to the Event Chat page for this event

14.6.6. An edit button

14.6.6.1. Directs the user to the Create Event Page for this event

14.7. The current user can attend events

14.7.1. Contains the number of people that are coming with the user

14.7.2. A view invitation button

14.7.2.1. Directs the user to the Invitation Page for this event

14.7.3. A button for the event chat

- 14.7.3.1. Directs the user to the Event Chat page for this event
- 14.7.4. An invitation button
 - 14.7.4.1. Directs the user to the Invitation Page for this event

15. Design Requirements

- 15.1. All pages utilize responsive design
 - 15.1.1. Menus will constantly be expanded when the screen is large
 - 15.1.1.1. Menus will always be accessed through smaller buttons when the screen is small
 - 15.1.2. The event list rows will include more events as the window width increases
- 15.2. Must function on Firefox

16. Database Requirements

- 16.1. The database will be relational
- 16.2. The database will store user account information
 - 16.2.1. The database will be able to retrieve user data
 - 16.2.2. The database will be able to update user data
 - 16.2.3. The database will be able to insert new user data
 - 16.2.4. The database will be able to delete user data
- 16.3. The database will encrypt user data

17. Optional Requirements

- 17.1. User can upload images to the database for invitation images
- 17.2. Back button

Updated Timeline

See Figure 3 in the appendix for a detailed description.

Appendix

Figure 1: Team Member Names

Name	Role
Benjamin Bruyns	Developer, Database
Peggy Lewis	Developer, Client
Spenser Morey	Developer, Server

Figure 2: AllEvently Tech-Stack

Component	Dependency Name	Dependency Type	Rationale
Front-End Framework	React.js	Core Framework	React is a highly efficient and popular JavaScript library for building interactive and dynamic user interfaces.
State Management	Context API	State Management	Handles complex states across the app (e.g., user authentication, events, guest lists) for predictable and scalable state management.
UI/Styling	Material-UI	UI/Styling Framework	Material-UI offers pre-built components CSS for fast and responsive design.
Form Handling	Formik + Yup	Form Library	Formik simplifies form creation and validation, while Yup provides schema-based validation to ensure form data integrity (e.g., login, sign-up, and event creation forms).
Google Maps	Google Maps JavaScript API	3rd-Party API	Enables Google Map integration for address verification, Places Autocomplete for event location suggestions, and geocoding for address validation.
Back-End Framework	Node.js + Express.js	Core Framework	Node.js allows for non-blocking, event-driven server-side code. Express.js is a minimalistic web framework that handles routes and middleware for APIs.
Authentication (Google)	Passport.js and Google Sign-In Library	Authentication	Passport.js integrates various authentication strategies (e.g., email/password, Google OAuth) and works well with Express for secure user authentication.
QR Code Generation	qrcode (npm package)	QR Code Generator	The qrcode npm package is a simple and beginner-friendly library for

			generating QR codes in Node.js, which can be used for event invitations or guest check-ins.
QR Code Display (Front-End)	react-qr-code (npm package)	QR Code Display	react-qr-code is a React component that generates QR codes on the front end, allowing users to scan event information or invitations via QR code.
Password Encryption	bcrypt.js	Encryption Library	They are used to hash passwords securely before storing them in the database to ensure user data security.
Database	PostgreSQL	Relational Database	PostgreSQL is a scalable, relational database that handles structured data like user accounts, events, and guest lists while supporting complex queries and data integrity.
Hosting Platform	Heroku	Cloud Hosting	Heroku allows for seamless deployment of full-stack applications with built-in support for Node.js.
Version Control	Git + GitHub	Version Control	GitHub provides a collaborative platform for version control, allowing team members to track changes, manage branches, and collaborate efficiently.
Unit Testing (Front-end)	Jest	Testing Framework	Jest is a widely used JavaScript testing framework suitable for unit testing front-end components and ensuring the correctness of UI elements.
Task Management	Trello	Project Management	Trello provides a visual way to organize project tasks, manage sprints, and track progress.

Figure 3: Tentative Schedule*

Date	Activity
<u>No Classes:</u> August 26 th : Day of Service Week 1: August 28 th – August 30 th	All Members: <ul style="list-style-type: none"> Syllabus review with tips and advice on successful software development Establish the project's main idea and goals
<u>No Classes:</u> September 2 nd : Labor Day	All Members: <ul style="list-style-type: none"> Present the Project Proposal Assess team members' strengths and abilities

Week 2: September 3 rd – September 6 th	<ul style="list-style-type: none"> Build the Requirements Document using a Flowchart Diagram Benjamin: <ul style="list-style-type: none"> Set up GitHub repository.
Week 3: September 9 th – September 13 th	All Members: <ul style="list-style-type: none"> Research tools and software for the project Continue gathering and establishing requirements
Week 4: September 16 th – September 20 th	All Members: <ul style="list-style-type: none"> Present the Requirements Document and Benjamin: <ul style="list-style-type: none"> Complete database setup for user accounts and authentication Peggy: <ul style="list-style-type: none"> Develop password reset and update password forms Spenser: <ul style="list-style-type: none"> Set up API for password reset, sign-up, and login validation
Week 5: September 23 rd – September 27 th	Benjamin: <ul style="list-style-type: none"> Set up database tables for events and guests Peggy: <ul style="list-style-type: none"> Create a front-end for a password reset, including email validation and error handling Spenser: <ul style="list-style-type: none"> Implement backend routes for password reset and account management
Week 6: September 30 th – October 4 th	All Member: <ul style="list-style-type: none"> Review Login, Sign-Up, and password reset flow Benjamin: <ul style="list-style-type: none"> Finalize database schema for event creation Peggy: <ul style="list-style-type: none"> Develop the Event Page UI (create event button, list of events). Spenser: <ul style="list-style-type: none"> Implement API for event creation and guest management
Week 7: October 7 th – October 11 th	All Members: <ul style="list-style-type: none"> Set up database structure for event invites and guest lists Peggy: <ul style="list-style-type: none"> Develop front-end for Event Creation Page Spenser: <ul style="list-style-type: none"> API development for guest invitations and event management
<u>No Classes:</u> October 14 th & 15 th : Fall Break Week 8: October 16 th – October 18 th	Benjamin: <ul style="list-style-type: none"> Finalize event creation and management features Peggy: <ul style="list-style-type: none"> Develop front-end for event chat and guest list Spenser:

	<ul style="list-style-type: none"> Implement event chat API
Week 9: October 21 st – October 25 th	Benjamin: <ul style="list-style-type: none"> Finalize database for event chat and guest list Peggy: <ul style="list-style-type: none"> Develop front-end for guest list management and invitation handling Spenser: <ul style="list-style-type: none"> API integration for event chat and guest list
Week 10: October 28 th – November 1 st	All Members: <ul style="list-style-type: none"> Testing and debugging of Login, Sign-Up, password reset, and event management components Benjamin: <ul style="list-style-type: none"> Implement database optimizations Peggy: <ul style="list-style-type: none"> Refine UI for guest list and event chat Spenser: <ul style="list-style-type: none"> Bug fixes in API routes and data handling
Week 11: November 4 th – November 8 th	All Members: <ul style="list-style-type: none"> Begin testing all components together. Solve issues arising from integration testing Benjamin: <ul style="list-style-type: none"> Performance checks on database queries
<p><u>No Classes:</u> November 11th: Veteran's Day</p> Week 12: November 12 th – November 15 th	All Members: <ul style="list-style-type: none"> Begin testing all base components (Login, Sign-up, Event Management) Solve any issues arising during the initial testing phase
Week 13: November 18 th – November 22 nd	All Members: <ul style="list-style-type: none"> Final review of event creation, guest list, and chat features. Fix any final issues Benjamin: <ul style="list-style-type: none"> Finalize database structures for optional features (image uploads, notifications)
Week 13: November 25 th – November 27 th <p><u>No Classes:</u> November 28th & 29th: Thanksgiving Break</p>	All Members: <ul style="list-style-type: none"> Review optional components (image uploads, back button, or follow-up notifications) Final bug fixes and polish
Week 14: December 2 nd – December 4 th <p><u>No Classes:</u> December 5th: Study Day</p>	All Members: <ul style="list-style-type: none"> Final touches on optional features and general polishing Prepare for project presentation Review the project for any final refinement