

AllEvently Report

Benjamin Bruyns, Peggy Lewis, Spenser Morey

Introduction

AllEvently is a versatile website application designed to simplify event management and enhance attendee engagement by addressing common challenges in event organization. It provides tools for creating events, inviting guests, and facilitating interaction among attendees, all in one platform. Integrating features like a live chat for real-time communication and multiple invitation options—digital through the app or printed posters—AllEvently solves the problem of fragmented event planning. Hosts can efficiently manage guest responses and logistics. At the same time, attendees enjoy a seamless way to RSVP, stay updated, and connect with others involved in the event, creating a more cohesive and interactive experience.

Technology

GitHub

GitHub is a platform for storing git repositories and ensuring version control. We chose this platform because it was a requirement for the CSCI 4805 Project.



Vercel

Vercel is a website and database hosting platform. Vercel was chosen because it promised to be easy to upload our git repository and deploy the website continuously. We also chose it because it promised to host a relational database, which we needed to create our website.



PostgreSQL

PostgreSQL is a framework for databases. We chose it because we needed a relational database, and Vercel offered hosting for PostgreSQL Database.



Vue.js

Vue.js is a framework for creating HTML for the user interface. We chose it because it advertised as an easy framework to use and implement.



Node.js

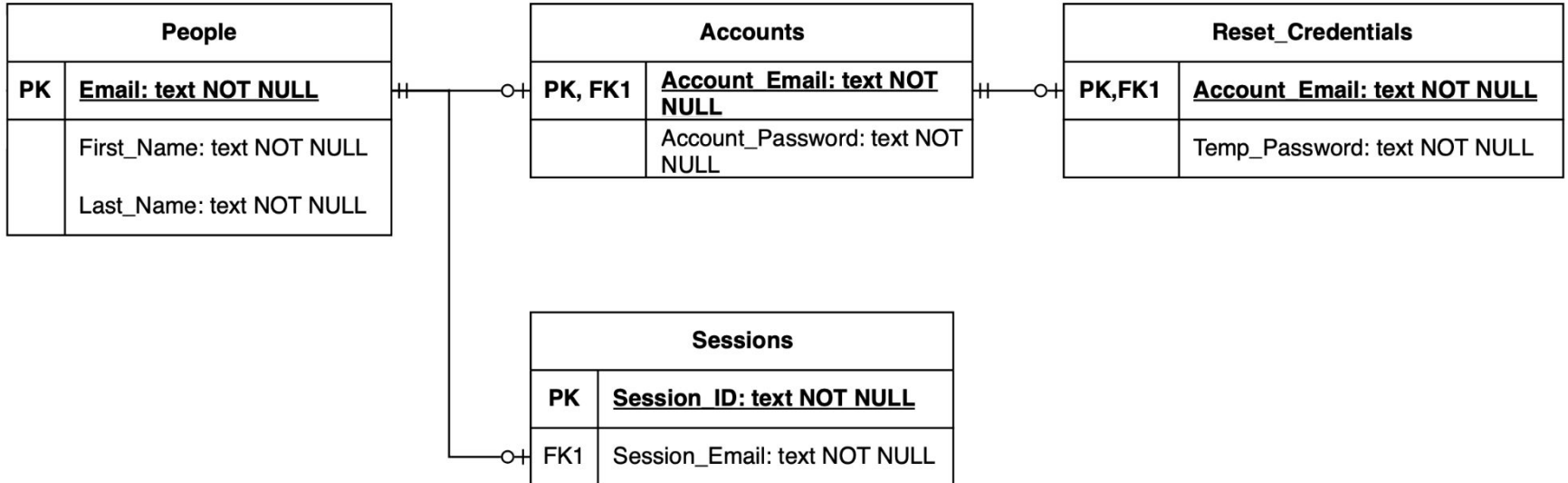
Node.js is a framework for creating the backend functionality of a website. We chose it because it is an easy framework to use and implement.



Database Design

User Tables

The collective purpose of these tables is to manage user data and interactions systematically, ensuring secure access, authentication, and account management within the website.



Events

This table keeps track of all events that have been created.

The foreign key references the People table.

Events	
PK	<u>EventID: int NOT NULL</u>
FK1	Event_Host: text NOT NULL Host_Name: text NOT NULL Event_Name: text default 'My Event' Event_Location: text NOT NULL Event_Start_Date: timestamp default currentdate()+1 Event_End_Date: timestamp Event_Time_Zone: text default 'UTC' Invitation_Layout: text default 'Center' Background_Image: int default 0 Font_Background_Color: text default '#000000' Font_Color: text default '#FFFFFF' Font: text default 'italic bold 20px arial serif' Is_Public: boolean default FALSE Reoccurs: boolean default FALSE Reoccur_Freq: int End_Reoccur: timestamp Request_Child_Num: boolean default FALSE Limit_Additional_Guests: boolean default FALSE Max_Additional_Guests: int default 0 Notify_Host: boolean default TRUE

Guests

This table keeps track of messages sent by users in each events.

Guests	
PK,FK1	<u>Event ID: int NOT NULL</u>
PK,FK2	<u>Guest Email: text NOT NULL</u>
	Responded: Boolean default FALSE Invite_Sent: Boolean default FALSE Accepted: Boolean Child_Count: int Guest_Count: int

Chat Messages

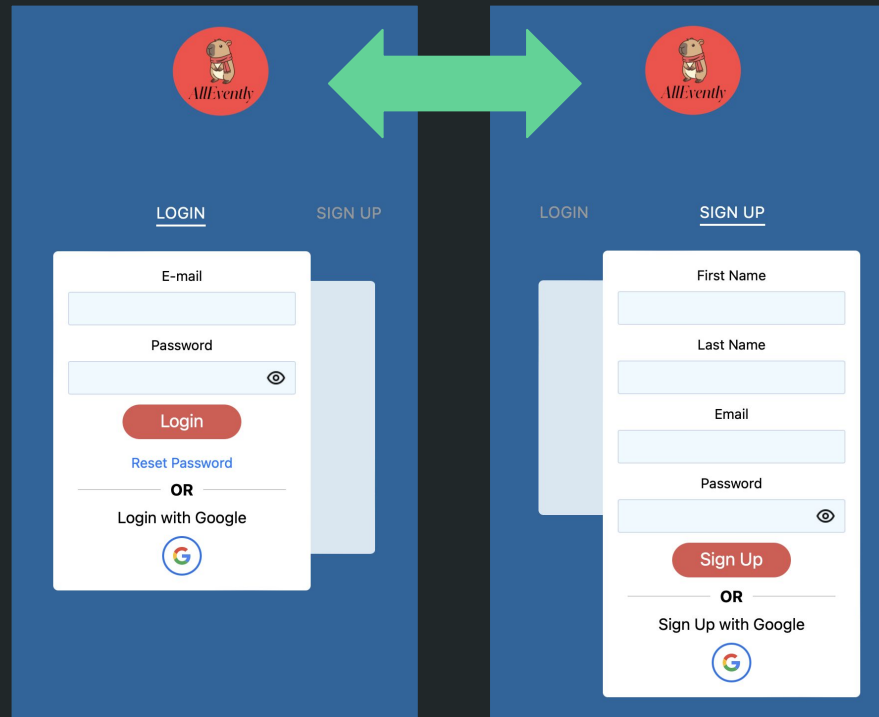
This table stores information about individuals who have signed into the website and those invited to events

Chat_Message	
PK	<u>Message ID: int</u>
FK1	Event_ID: int NOT NULL
FK2	Sender_Email: text NOT NULL
	Content: text NOT NULL
	Date: timestamp default currentdate()
	Accepted: boolean

Frontend Design

Login/Signup


This page that shows the user a login and a sign-in dialogue you can switch between. Both login and sign-in have access to the GoogleSSO.vue display, allowing the user to use Google Single-Sign-On to access our website.



Password Changing

The Password Reset page is for sending the reset code to the user.

The Password update page allows a user to change the password using the code.




Password Reset

Email:

Reset Password

[Cancel](#)



Update Password

Emailed Code:

New Password:

Confirm Password:

Update Password

[Cancel](#)

Events

A page that shows users all events they attend as EventCard.vue displays


Events

[Attending](#)[Hosting](#)

Upcoming Events

Lisa's Baby Shower (Public event)

Hosted By: Guest



Sep 28, 2024
3:00 PM CDT

Wilma Rudolph Event Center
8 Champion's Wy, Clarksville, TN 37040

[Edit Invitation](#)[Guest List](#)[Print](#)[Event Chat](#)

Public Events

A page that shows users
all public events on
PublicEventCard.vue
displays

Jonie's Birthday (Public event) Hosted By: Alice



Oct 5, 2024

6:00 PM CDT

Smith-Trahern Mansion

101 McClure Street, Clarksville, TN 37040

[View
Invitation](#)

[Event Chat](#)

Carson's Bounce Party

(Public
event)

**Hosted By:
Sue**



Oct 21, 2024

12:00 PM

Sky Zone

310 Needmore Rd, Clarksville, TN 37040

[View
Invitation](#)

[Event Chat](#)

Guest List

A page that shows event hosts the RSVP status of all invitees.

Guest List

Attending: 1 Waiting: 3 Regrets: 0

John Smith

Invite Sent

View

Ashley Williams

Not Sent

View

Hannah Barbara

Not Sent

View

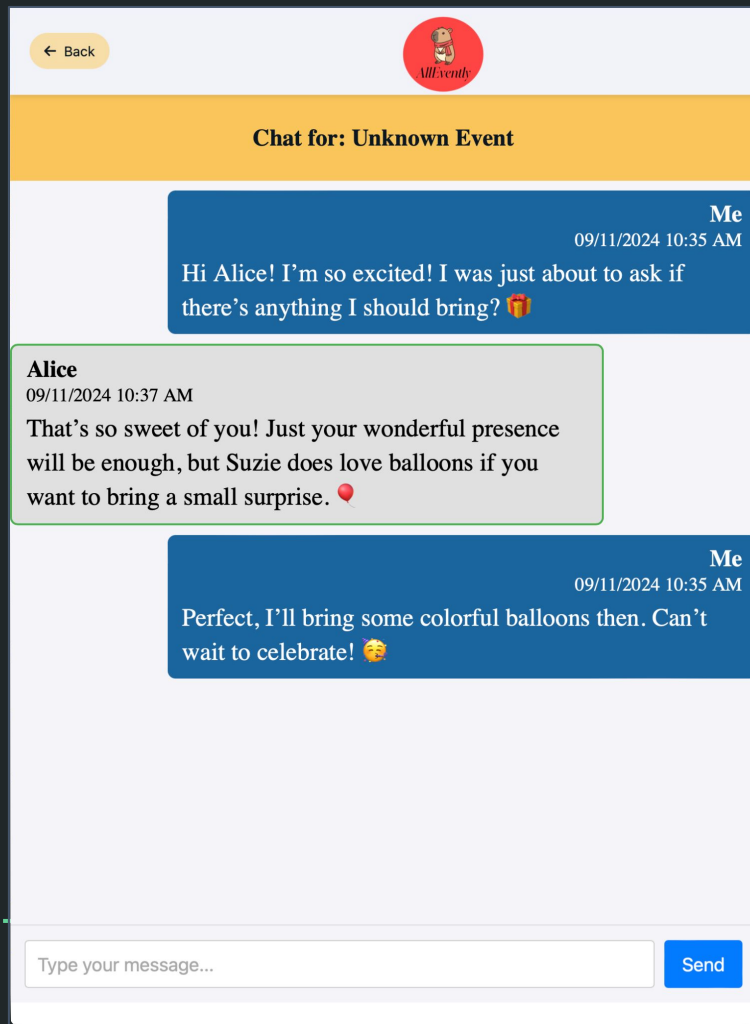
Wilhelm Schultz

Not Sent

View

Chat

A page that shows users the messages sent in a given event.



Backend Design

Account Handling

The backend has two functions for managing user information in the database.

- `signup.ts` for creating user accounts.
- `authentication.ts` for checking login credentials.
- `currentuser.ts` for fetching information of the current user is.

Event Management

The backend has 3 functions for managing events from the database.

- `attendingevents.ts` for fetching events you have been invited to or have RSVPed.
- `hostedevents.ts` for fetching events that you are hosting.
- `publicevents.ts` for fetching events that are public.
- `eventcreation.ts` for creating new events.

Vercel Deploy/Build Instructions

Fork Git Repo

Go to <https://github.com/BenStorm2514/AllEvently>

The screenshot shows the GitHub repository page for **AllEvently** by user **BenStorm2514**. The repository is public and has 0 forks and 0 stars. The **Fork** button is highlighted with a red box and a red arrow pointing to it. The repository has 1 branch (main) and 1 tag. The commit history shows a merge of the remote-tracking branch 'origin/main' by **plewis10apsu** 10 hours ago, with 257 commits. The file list includes **AllEventlyProjectFiles**, **Documents**, **.gitignore**, **README.md**, and **package-lock.json**. The **README** section is visible at the bottom, showing the repository name **AllEvently**.

BenStorm2514 / AllEvently

Search: Type / to search

Code Issues Pull requests Actions Projects Wiki Security 1 Insights Settings

AllEvently (Public)

Pin Unwatch 1 Fork 0 Star 0

main 1 Branch 1 Tag

Go to file Add file Code

plewis10apsu Merge remote-tracking branch 'origin/main' ✓ 570e649 · 10 hours ago 257 Commits

File	Commit Message	Time
AllEventlyProjectFiles	Merge remote-tracking branch 'origin/main'	10 hours ago
Documents	Update database and cheats sheet	14 hours ago
.gitignore	Merge remote-tracking branch 'origin/main'	2 days ago
README.md	Update README.md	3 weeks ago
package-lock.json	Fixed issues with casting to strings	2 days ago

README

AllEvently

About

all-evently.vercel.app

Readme

Activity

0 stars

1 watching

0 forks

Releases

1 tags


Create a new release

Packages

No packages published

Publish your first package

Create Vercel Account



Log InContact

Your first deploy
is just a sign-up away.

Plan Type


☒ I'm working on personal projectsHobby

☐ I'm working on commercial projectsPro

Your Name

Continue

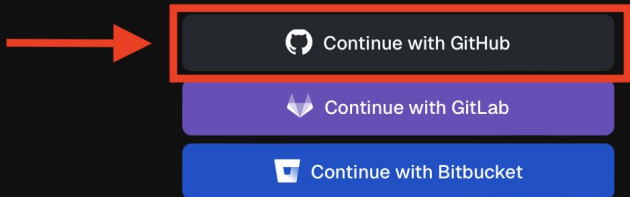
By joining, you agree to our [Terms of Service](#) and [Privacy Policy](#)

Have a complex company use case? Get Enterprise grade assistance 

Create Vercel Project

Import Git Repository

Select a Git provider to import an existing project from a Git Repository.



After continuing with GitHub.
Choose to connect to your
Github Account and connect to
your AllEvently fork.

Set Deployment parameters

Make sure that you set the deployment parameters as shown in the red boxes.

New Project

Importing from GitHub

BenStorm2514/AllEvently main AllEventlyProjectFiles/frontend

Choose where you want to create the project and give it a name.

Vercel Team

Project Name

benstorm2514's projects Hobby

all-evently-119n

Framework Preset

Vite

Root Directory

AllEventlyProjectFiles/frontend

Edit

Build and Output Settings

Build Command

@types/node && npm run build



Output Directory

dist



Install Command

'yarn install', 'pnpm install', 'npm install', or 'bun install'



Google Maps API

Go to the Google Cloud Console and log in with your Google account. Create a new project by clicking the project dropdown at the top and selecting New Project. Name the project, like "AllEvently Project," and create it. Once the project is created, go to APIs & Services > Library in the left sidebar and enable the Maps JavaScript API, Places API, and Static Maps API. After enabling the APIs, go to APIs & Services > Credentials and click Create Credentials > API Key. Copy the generated API key and edit it to restrict its usage. Under Application Restrictions, choose HTTP Referrers (websites) and add http://localhost/* for local development and https://your-vercel-domain.vercel.app/* for production. Also, restrict the key to the APIs enabled earlier and save the changes.

Add Google API Key

Environment Variables

In order to provide your Deployment with Environment Variables at Build and Runtime, you may enter them right here, for the Environment of your choice. [Learn more](#)

A new Deployment is required for your changes to take effect.

[Create new](#) [Link Shared Environment Variables](#)

Sensitive

If enabled, you and your team will not be able to read the values after creation. [Learn more](#)

☐ Disabled

Environments

☒ Production

☒ Preview [Select custom branch](#)

☒ Development

Key	Value
VITE_GOOGLE_MAPS_API_KEY	API_KEY_VALUE_HERE

[+ Add Another](#)

[Import .env](#) or paste the .env contents above

[Save](#)

After continuing with GitHub.
Choose to connect to your
Github Account and connect to
your AllEvently fork.

Add Google API Key

Go back to your Vercel project and go to Settings > Environment variables and make a new Key.

Make sure to save.

Environment Variables

In order to provide your Deployment with Environment Variables at Build and Runtime, you may enter them right here, for the Environment of your choice. [Learn more](#)

A new Deployment is required for your changes to take effect.

[Create new](#) [Link Shared Environment Variables](#)

Sensitive

If enabled, you and your team will not be able to read the values after creation. [Learn more](#)

☐ Disabled

Environments

☒ Production

☒ Preview [Select custom branch](#)

☒ Development

Key	Value
VITE_GOOGLE_MAPS_API_KEY	API_KEY_VALUE_HERE

[Add Another](#)

[Import .env](#) or paste the .env contents above

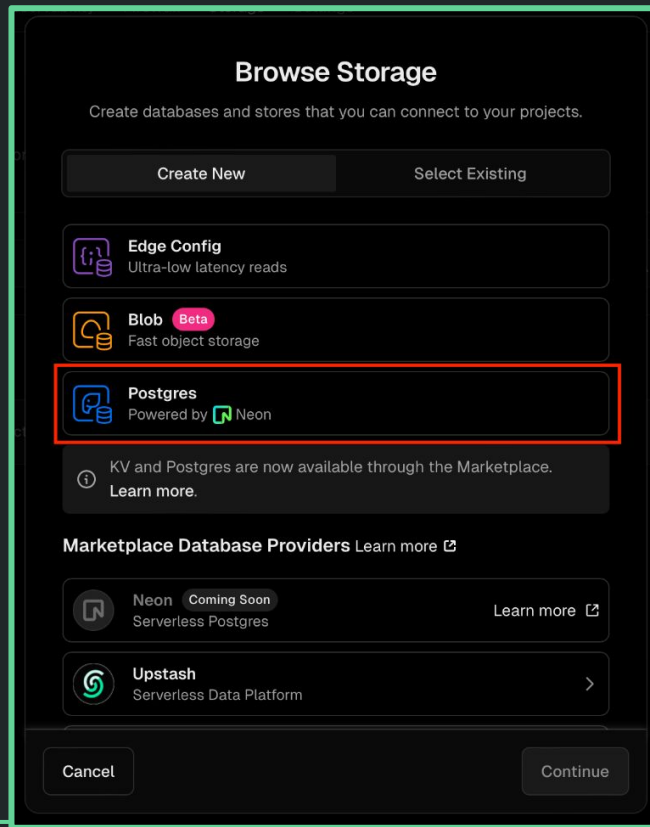
[Save](#)

Database Deploy/Build Instructions

Create Vercel Database

Go to the Storage tab in your project and press Create Database.

Select Postgres.




Connect Vercel Database

Go to the project tab in the database and press connect project and select your project.

Projects

These are the projects that are connected to this database.

Connect Project

<input type="checkbox"/>	Project	Environment Variables	Environments
<input type="checkbox"/>	 all-evently	POSTGRES_URL POSTGRES_URL_NON_POOLING POSTGRES_URL_NO_SSL POSTGRES_PRISMA_URL POSTGRES_USER POSTGRES_PASSWORD POSTGRES_HOST POSTGRES_DATABASE	Production, Preview, Development
<input type="checkbox"/>	 all-evently-backend	POSTGRES_URL POSTGRES_URL_NON_POOLING POSTGRES_URL_NO_SSL POSTGRES_PRISMA_URL POSTGRES_USER POSTGRES_PASSWORD POSTGRES_HOST POSTGRES_DATABASE	Production, Preview, Development

Create Database Functionality

Next, go to the file git repo and go to Documents->Database->AllEventlyDB.sql. Copy the contents of this file and go back to the Vercel database. Go to the query tab, paste the SQL code into the prompt, and run the query.

Thank You!