

LHS Job Shadow Technical Documentation
Written By Anish Lakkapragada '24

Introduction.....	5
This Guide.....	5
Must Read Sections Of This Guide.....	5
Platform Overview.....	6
Who To Contact.....	6
Next.js Frontend.....	6
Overview.....	6
Explanation of each Page Route.....	7
/.....	7
/login.....	7
/signup.....	7
/reset.....	7
/viewHosts.....	8
/viewAllHosts.....	8
/missingHosts.....	8
/student/[studentId].....	8
/student/[studentId]/[eventId].....	9
/student/[studentId]/add.....	9
/recruiter/[userId].....	9
/recruiter/[userId]/manageEvent.....	9
/recruiter/[userId]/addh.....	9
/recruiter/[userId]/add.....	10
/recruiter/[userId]/hosts/[hostId].....	10
/recruiter/[userId]/hosts/[hostId]/editForm.....	10
/admin/[slug].....	10
/admin/[slug]/[eventId].....	10
/admin/[slug]/[eventId]/data.....	12
/admin/[slug]/[eventId]/confirmations.....	12
/admin/[slug]/[eventId]/hostprefs.....	13
/admin/[slug]/[eventId]/importante.....	13
/admin/[slug]/delete.....	13
/admin/manage.....	13
Components.....	13
DataLoad.tsx.....	13
EditEventForm.tsx.....	14
EditHostForm.tsx.....	14
EventCard.tsx.....	14
StudentEventCard.tsx.....	14
StudentEditGrade.tsx.....	14
PublicHostCard.tsx.....	14

AdminRecruiterHostCard.tsx.....	14
StatusPortal.tsx.....	15
Approve.tsx, Event.tsx.....	15
Database.....	15
Overview.....	15
Schema Overview.....	15
User.....	16
Student.....	16
OpportunityProvider.....	17
JobShadowEvent.....	17
StudentMember.....	17
RecruiterEventPair.....	17
Host.....	18
AssignedHosts.....	18
HostPreferences.....	18
MatchedStudent.....	18
RejectedMatchedStudents.....	19
AuxiliaryPerson.....	19
Summary.....	19
User Authentication.....	19
Next.js Backend API.....	20
Overview.....	20
Explanation of Individual API Route.....	20
/.....	20
/allDBEvents.....	20
/missingHosts.....	20
/getAllHosts.tsx.....	20
/users/getAll.....	20
/users/delete.....	21
/users/get.....	21
/student/.....	21
/student/add.....	21
/student/getStudent.....	21
/student/email_redirect.ts.....	21
/student/status.ts.....	21
/student/events/[eventId]/.....	21
/student/events/[eventId]/changeAttendedMtg.....	22
/student/events/[eventId]/changeSeeAssignment.....	22
/student/events/[eventId]/decideHost.....	22
/student/[studentId]/getHostsAndPrefs.ts.....	22
/recruiter/.....	22

/recruiter/getRecruiter.....	22
/recruiter/add.....	22
/recruiter/add_host.....	22
/recruiter/events/approved.....	23
/recruiter/events/.....	23
/recruiter/[recruiterId]/joinEvent.....	23
/recruiter/[recruiterId]/leaveEvent.....	23
/recruiter/[recruiterId]/hosts/add.....	23
/recruiter/[recruiterId]/hosts/delete.....	23
/recruiter/[recruiterId]/hosts/edit.....	23
/recruiter/[recruiterId]/hosts/get.....	23
/recruiter/[recruiterId]/hosts/getEvents.....	24
/recruiter/[recruiterId]/hosts/remove.....	24
/recruiter/[recruiterId]/hosts/removeSpec.....	24
/events.....	24
/events/students/[eventId].....	24
/events/students/[eventId]/unverified.....	24
/events/students/[eventId]/resetStudent.....	24
/events/students/[eventId]/getHostPrefs.....	25
/events/hosts/[eventId]?approved=true/false.....	25
/events/hosts/[eventId]/approved.....	25
/events/hosts/[eventId]/sendEmail.....	25
/events/[eventId]/.....	25
/events/[eventId]/confirmations.....	25
/events/[eventId]/getAllRecruitersAndHosts.....	25
/events/[eventId]/importantInfo.....	26
/admin/.....	26
/admin/add.....	26
/admin/[eventId]/assignHostPreferences.....	26
/admin/[eventId]/edit.....	26
/admin/[eventId]/getAssignments.....	26
/admin/[eventId]/transfer-host.....	26
/admin/[eventId]/host-approval.....	27
/admin/[eventId]/student-approval.....	27
/admin/[eventId]/student-form.....	27
/admin/[eventId]/student-grade-update.....	27
/admin/[eventId]/recruiter-approval.....	27
/admin/[eventId]/recruiter.....	27
/admin/[eventId]/manualAssign.....	27
/admin/[eventId]/runthealgorithm.....	27
/admin/[eventId]/calibratecounts.....	28

/admin/[eventId]/calibratestudents.....	28
/admin/[eventId]/unlink/host.....	28
/admin/[eventId]/unlink/student.....	28
What Needs to Be Done At Each Step of Job Shadow.....	28
Before Job Shadow Student Registration.....	28
During Student Registration.....	28
In the case that a student loses their password:.....	28
During The Week When Students Submit Their Preferences.....	29
After Job Shadow Day.....	29
Services, Credentials, Plans, and Limits.....	29
Supabase – PostgreSQL Database & User Authentication.....	29
Plans/Limits.....	29
Credentials.....	29
Vercel – Web Hosting.....	29
Plans/Limits.....	30
Credentials.....	30
GoDaddy – Domain Management.....	30
Plans/Limits.....	30
Credentials.....	30
Resend – Email Server Management.....	30
Plans/Limits.....	30
Credentials.....	31
Bugs/Future Implementations.....	31
Bugs.....	31
Distributing this platform to other organizations.....	31
.env Credentials & Service Passwords.....	32
Service Passwords.....	32

Introduction

This Guide

This is a guide for managing the entire [LHS Job Shadow Platform](#) before, during, and after the annual Lynbrook Job Shadow cycle. This guide is intended for the LHS Job Shadow Tech Team. This guide was written after the LHS Job Shadow 2024 Program, which was the first Job Shadow program since 2019 and used a new [platform](#).

Please circulate copies only to those who need it. A PDF of this guide should always be available at <https://anish.lakkapragada.com/notes/lhsjobshadow.pdf>.

Must Read Sections Of This Guide

This guide is long. But there are certain sections worth reading:

- This section
- /admin/[slug]/[eventId] route in the Next.js Frontend Section
- StatusPortal.tsx component in the Next.js Frontend Section
- The Database Schema Section
- The /admin/[eventId] routes in the Next.js Backend Section
- What Needs To Be Done At Each Step of Job Shadow
- Services, Credentials, Plans, and Limits Section
- Bugs/Future Implementations Sections
- .env Credentials & Service Passwords Section

Platform Overview

There are 6 different components to this website. They are listed below:

- Next.js Frontend
- Database
- Next.js Backend
- Site & Domain Management
- Email Management

Each of these different components is detailed in this guide. Additionally, a section is added to detail what is done at every step of the Job Shadow program and bug fixes/implementations/security vulnerabilities. That being said, it is highly recommended to read through the Next.js Backend and Database section to understand the entire complexity of the system and unintuitive design quirks.

Prerequisites

This guide assumes an understanding of what the LHS Job Shadow Program is, the different parties involved in the program, and the different phases of the program.

Furthermore, the languages and technologies used are: Next.js, TypeScript, Vercel, Supabase, PostgreSQL, TailwindCSS.

Who To Contact

Please contact Anish Lakkapragada (anish.lakkapragada@gmail.com, 408-409-9673) if you encounter any issues that the Lead Developer / team cannot solve.

Next.js Frontend

Overview

The core of the website is the actual frontend. The frontend is built with Next.js 13. The UI of the site is given with the Mantine UI Library (version ^6.0.13) and provided with TailwindCSS. The scripts to run to start the app locally are in the **package.json**, although they should be pretty self-explanatory.

The Next.js frontend is provided in the `app/` folder. Note that this guide does not cover layout files as they are too simple.

Explanation of each Page Route

The following description describes what each individual page route does. Each route is a Typescript Extended (.tsx) file.

/

Default home page of the platform. Shows important information on dates, surveys, descriptions and should always be up.

/login

Page for users to login based on their email.

/signup

Provides a page for users of the site to register. Users receive different forms depending on the user type they choose (student, recruiter, admin.) **All** user signups will receive an email from Supabase which they have to click on in order to verify their account. They will **not** be able to login without this email.

This page may be found in the `/admin/[slug]/page.tsx` file instead, as it may be requested by LHS Job Shadow administrators to move it to a less-accessible link to prevent any further signups from students.

Note that for students, they will by **default** after sign-up join the current LHS Job Shadow event. However, by default they are not approved to join this event. This event ID is given in line 71 of the `StudentForm.tsx` component.

/reset

Allows a person to try to reset their password by sending an email to their account.

/viewHosts

Shows information for the hosts which **have been approved** for the given LHS Job Shadow Event (change the event id with the EVENT_ID variable that is hardcoded.)

/viewAllHosts

Shows information for the hosts **regardless of approval status** for the given LHS Job Shadow Event (change the event id with the EVENT_ID variable that is hardcoded.)

/missingHosts

Gives a list of all the hosts that have been entered into the Database but are missing from the LHS Job Shadow event.

/student/[studentId]

This page is the default landing page for a student after they sign-in. Please note that the [studentId] parameter is NOT their numerical studentId, but instead the UUID that is used in the Database to identify them (see Database section for more details.)

This page shows the names of the Job Shadow events this student is participating in and gives them a button for each event to click on that will take them to a page to submit their host preferences for the event (/student/[studentId]/[eventId]). **However, if they are currently NOT approved for this event, that button will be disabled. The button is also disabled if they have already submitted their preferences for that event. The button is also disabled if the time has not come for the student's grade level to register.**

There is also, below, a Portal Section for the student to view updates on the Job Shadow Event. These updates would include:

- Being matched in an assignment run to an employer
 - Requesting Accept/Reject of a student based on the job they were assigned to
- Telling a student they have not been matched to any event
- Reminding a student who has accepted a job that they have to go to mandatory orientation on a given date

- Showing a student who has gone to the mandatory orientation all the information about their host

Thus, this is a very critical page.

`/student/[studentId]/[eventId]`

A student should only be able to access this page if they are approved for this event with the eventId in the given URL/page route AND if they have not submitted the form this page has before AND if it is their time for their grade level to submit this form to give their host preferences.

On the page, students are able to submit their Top 10 host preferences. Students must only select one to submit and of course cannot choose the same host for two different preferences. They also cannot leave an empty preference (e.g. give one host for preference 1 and one host for preference 3, but none for preference 2).

`/student/[studentId]/add`

This is a **hidden** page. It is meant to allow a student to add themselves to any Job Shadow event in the platform and is strictly meant for testing purposes (e.g. wanting to add yourself to an event you created.)

If you are planning to try to distribute this software beyond Lynbrook High School, instead clone the repository and replicate the platform instead of using this approach.

`/recruiter/[userId]`

This is the default landing page for recruiters after they login. On this page, they are able to view cards showing all of the hosts they have created.

`/recruiter/[userId]/manageEvent`

This is the page for recruiters to basically allow themselves to join a given event, regardless of whether they have created any hosts already. This form essentially creates the link between the Recruiter and the event (RecruiterEventPair row in the database.) By joining an event, recruiters enable themselves to be visible to the event administrators.

`/recruiter/[userId]/addh`

This page provides a form for recruiters to create a new host. Here, they enter all of the information for a host.

`/recruiter/[userId]/add`

This page provides a form for recruiters to add a host they have created to a certain event via a dropdown menu.

`/recruiter/[userId]/hosts/[hostId]`

This page shows information on a given host that a recruiter has created. Here they can edit the host information themselves, view the events this host is participating in, and “Reset Host Edit Form Chances.” What this does is give 2 more opportunities for the actual host point-of-contact to submit the form to finalize the host information. This submitted form is detailed below. To copy the link of this form for the actual host point-of-contact, a button is provided as well.

`/recruiter/[userId]/hosts/[hostId]/editForm`

Form for the host point-of-contact to be able to finalize all information on their host. They have by default only 2 chances to submit this form. The recruiter can reset it to 0 to give the host point-of-contact more chances (or the recruiter can, on the `/recruiter/[userId]/hosts/[hostId]` side, edit the form on behalf of the host point-of-contact.)

`/admin/[slug]`

This is the default landing page for an administrator. `[slug]` gives the user ID of the administrator account (found in the User table.) Was named `[slug]` based on a tutorial and has never been changed since.

On this page, the administrator can view all of their events with event cards.

`/admin/[slug]/[eventId]`

This is the **most** important page in the entire platform. On this page, the administrators can do a bunch of different tasks to manage hosts, recruiters, and students. They are detailed below.

- View Hosts Button
 - Admin can view all info about a host
 - Admin can edit a host
 - Admin can unlink a host from the event

- Admin can send email to the host point-of-contact about all the students who have been assigned to this host, and approve/unapprove a host
- View Students Button
 - Admin can view all info about student
 - Admin can approve/unapprove a student
 - Admin can change the grade of a student
 - Admin can track if the student has submitted the required form (i.e. the health/liability paper form to the front office)
 - Admin can “Reset Student Preferences” for a student
 - This clears the entirety of what hosts the student has been matched to, what host preferences they have entered, and just allows them to start from a clean slate
 - Admin can unlink a student (remove StudentMember connection)
- View Recruiters Button
 - Admin can view all the recruiters in the event and see all the hosts each recruiter manages.
- View Data Exports Page
 - Admin can be able to perform a lot of data exports by going to the Data Exports page (/admin/[slug]/[eventId]/data route).
- View Confirmations
 - Admin can be able to view the state of all students who have submitted the form (e.g. unassigned, matched & unconfirmed, confirmed, refused) in the /admin/[slug]/[eventId]/confirmations)
- View Missing Hosts
 - Admin can view the list of hosts which are in the database but are missing from a hardcoded event. Check /missingHost route in Next.js Frontend section for more details.
- View Host Preferences for a Student
 - Admin can be redirected to a page (/admin/[slug]/[eventId]/hostprefs) to view the host preferences a given student has entered.
- View Special Requirement Assignments
 - Admin can be redirected to a page (/admin/[slug]/[eventId]/importante) to view all students who are going to an event with special requirements or that is on a non-typical date.
- Edit This Event
 - Admin can open up a model to be able to edit this given event.

There are other functions as well!

On this page, the administrators can:

- **Run the assignment algorithm to Assign Students Based on their Host Preferences**
- Transfer Hosts between two recruiters
- Manually Assign Host Preferences for a given student in the event to be able to add a student to a given job shadow run, or to perform testing.

Note that because of how much data this page requires, it's preferred not to add any more functionality to it and instead just create another route 😊

`/admin/[slug]/[eventId]/data`

This page is the Data Exports page that essentially exports entire tables into .CSV files for administrators to manage with Excel.

Here's all the data exports provided on this page:

- Download Student Signups CSV
 - Gives information for all students who have joined the given event.
- Download Students with UNVERIFIED emails.
 - Gives information for all students who have emails which are unverified. **This particular data source takes the longest to load.**
- Download CSV of Student Assignments
 - Downloads information of all the students who have been assigned to a given job.
- Download CSV of Students Assigned To Hosts with Special Requirements
 - A Host has a special requirement if it has a non-null entry for the "importantInfo" field
- Download CSV of Students Assigned to Hosts on An Alternate Day
 - This refers to hosts whose times are not on the typicalDate for the event
- Download CSV of ALL Hosts in the Database.

More can be added very easily with the *react-csv* package.

`/admin/[slug]/[eventId]/confirmations`

This is another extremely important route. It shows the status of students who have **submitted their host preferences**. This includes students who are unassigned (not assigned to a host yet in an assignment run), students who are unconfirmed (have been assigned, still waiting on a response), students who have refused a match, and students who are confirmed (have accepted a match.) The functionality for the three groups provided in this route is detailed below.

1. Unassigned Students

Displays all students who aren't assigned to anything and their preference list. Also provides an option to enable manual assignments of an unassigned student to a host.

2. Unconfirmed Students

For all unconfirmed students, this page offers the option to make an assignment visible (page shows the time when this was enabled for students who can see their assignment) and provides the option for the administrators to accept/reject the host a student was matched to on their behalf.

3. Confirmed Students

Allows administrators to see what host a student has confirmed themselves to. Additionally, admin can “check-off” students for attending the mandatory orientation.

4. Refused Students

Lists all the students who have refused a host. List the host the student has refused.

`/admin/[slug]/[eventId]/hostprefs`

This allows an event administrator to see all the host preferences a student submitted based on their name.

`/admin/[slug]/[eventId]/importante`

This page allows administrators to see all special assignment cases for student assignments. The page is divided into two sections:

1. Listing all assignments of students to hosts with special requirements (i.e. the host has a non-null “importantInfo” field.)
 - a. Lists special requirement (“importantInfo” field) for the host.
2. Listing all assignments of students to hosts that are on an atypical date (i.e. the times the host is hosting students are not on the “typicalDate” field of the event.)
 - a. Lists date of job shadow for these hosts.

`/admin/[slug]/delete`

This is a route for an administrator to be able to delete ANY account (based on email) from the database. This route is commonly used to completely delete students from the database to allow them to re-sign up.

`/admin/manage`

Useless route, delete.

Components

In this section, I detail all of the Components that the Job Shadow Platform used. They can be found in the `components/` folder.

DataLoad.tsx

Component used to provide an interim “Data is loading ...” message.

EditEventForm.tsx

Form used by event administrators to be able to edit information about their event.

EditHostForm.tsx

Form used by recruiters to be able to edit information about their host.

EventCard.tsx

Cards displayed to admin on the /admin/[slug] page to allow them to click to view a given event they manage.

StudentEventCard.tsx

Cards displayed to students on the /student/[studentId] page to allow them to click to view a given event they are participating on to access the host preference submission form. This is only if they are approved for the event AND have not submitted their host preferences yet.

StudentEditGrade.tsx

Component to allow event admin to be able to edit the grade (i.e. change grade level for special student to 13) of a student.

PublicHostCard.tsx

Provides the public information about a given host. This is specifically to be shown on the public page /viewHosts.

AdminRecruiterHostCard.tsx

Provides a view of all the host information (more info than PublicHostCard.tsx card) to admin/recruiter user groups. If the user is admin, this card comes with options:

- to approve the host
- unlink the host (i.e. delete the relation of the host with the event in the AssignedHosts table)
- Send an email to the host point-of-contact to notify the host of all the students who have been assigned to this host (this email is sent PRIOR the job shadow date)
- Edit Host Information

StatusPortal.tsx

The section of the /student/[studentId] which is the “Status Portal” to notify the student of any updates in the Job Shadow events they are registered in. These are computed based on the StudentMember values for the Student (i.e. have the students submitted, have they been matched, have they accepted/rejected, etc.).

Based on these values, the Student Portal which show:

- A panel to accept/reject a job assignment
- A reminder for the student to go to mandatory orientation, with a confirmation message of the host they have accepted
- Show all host info if the student has gone to mandatory orientation
- Tell the student they have rejected a given host in the past

This is by far the most complex component.

Approve.tsx, Event.tsx

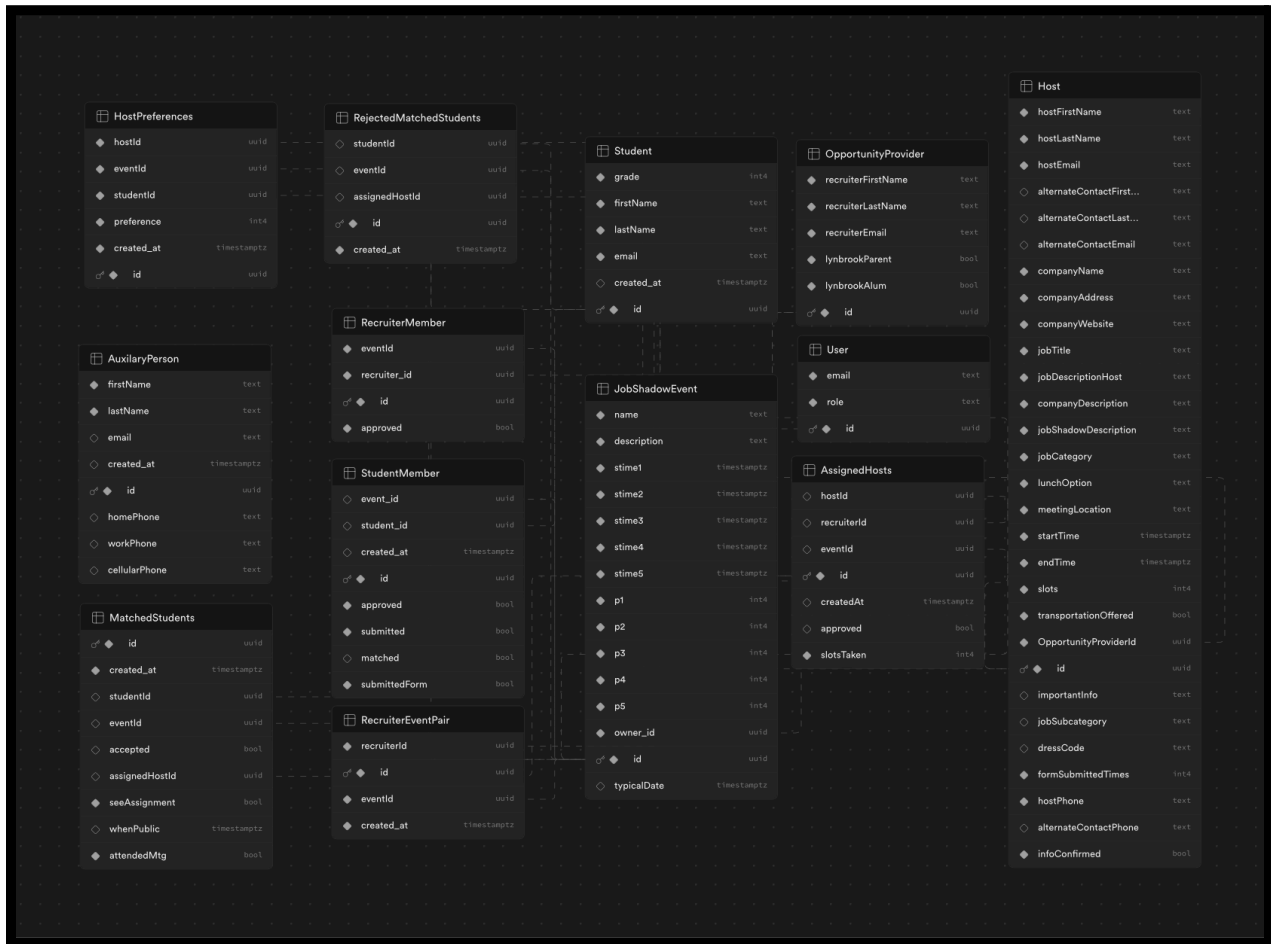
Useless components, please delete.

Database

Overview

The database for this project is hosted by a Supabase project. There are a total of 10ish tables. The schema for this project is managed through Prisma (can be accessed in the prisma/schema.prisma) file. The Prisma ORM is also used to perform CRUD operations on the table (see Next.js backend API routes.) Supabase, the Database Provider, also manages User Authentication.

Schema Overview



This is the Schema Visualizer for the Supabase tables.

The below sections detail information for each table. Note that every single table has an “id” field, which is of the UUID parameter. **Also note that every table may have a unique constraint, which can be found in the prisma.schema file.**

User

Stores information on a user whenever they register and sign up on the platform. Also stores the user type of a given user (“STUDENT”, “RECRUITER”, “ADMIN”). Note that the ID of the User is the same as the ID of the user in Supabase Auth’s internal auth.users table.

Student

Contains information on a given student.

Please note that grade level can be set to 13. This is meant to allow the student to be able to sign up extra early. This is for **special-cases**, such as when a student is the child of a recruiter or the student is on the LHS Job Shadow Tech Team :)

OpportunityProvider

Contains information on a given recruiter. Named OpportunityProvider. Information on Lynbrook Alumni and Student ID of child (if any that go to Lynbrook) is meant to give prioritization.

JobShadowEvent

Stores all information for a given created Job Shadow Event. In particular the p1, p2 ... p5 and stime1, stime2, ...stime5 fields are of interest. The p1 field represents the grade level which is the first priority (e.g. if p1=9, 9th grade is first priority). The stime1 field represents the time in which students of that grade level can be able to submit their host preferences.

The reason for the 5 priority timings is to account for a special grade, called the 13th grade level. This is for LHS Job Shadow Tech Team people who are participating in that year's Job Shadow program and students of recruiters who helped recruit companies/hosts for that year's Job Shadow program. The admin can temporarily change a student's grade to 13 so that they can be able to be included in this matching which happens earlier.

The typicalDate field is meant to give the date that **most** hosts for this event will have their job shadow day (March 18th for Job Shadow '24 program.) Using this date, admin can be aware of all students assigned to hosts which are not on this `typicalDate`.

StudentMember

This form stores the connection between a student and an event that they have joined (and thus are a "member" of.)

Three boolean fields of interest (all with default false; the administrators of the event in a StudentMember row can change these properties):

- approved – is this student approved to participate in this Job Shadow Event
- submittedForm – did this student submit their form to participate in this Job Shadow Event (i.e. the health form)
- submitted – did this student submit their host preferences for this Job Shadow Event already

- matched – has the student been matched once with a given host for this event. Note that if a student has decided to reject a given host they have been matched with, **this field is still true**.

RecruiterEventPair

Stores the link between a recruiter (OpportunityProvider) and an event that they have joined.

Host

Contains information on a Host. The “OpportunityProviderId” field provides the id of the OpportunityProvider who created this host.

The `slots` field stores the maximum number of slots that this Host is providing. For example, if a given host is hosting a maximum of 5 students, then this field would be equal to 5.

AssignedHosts

This parameter stores the connection between a host and its assignment to a given event (made possible through the /recruiter/[userID]/add page.) The approved field is a boolean that stores if the admin for the event has approved having this host (default false). The `slotsTaken` field stores the current number of slots for this host that have been taken by students who are matched to this host (this information is in the MatchedStudents table.) This `slotsTaken` field should **never** be higher than the `slots` field of the host.

HostPreferences

Stores a given preference (i.e. a number from 1-10) that a student has entered for a host provided by a given Job Shadow Event.

MatchedStudent

Stores the entries for StudentMembers who have been matched to a host in a given job shadow event. The host a given student has been matched to is given by the “assignedHostId”.

Some fields of interest:

- accepted – boolean field on whether the student has accepted/committed to this host they have been matched with. **Default NULL value.**
 - The value of the accepted field is **either true or null (null means we are waiting on their response). Read the highlighted section below to understand why.**
- attendedMtg – boolean field on whether the student has attended the mandatory orientation meeting. Default False.

- Note that only **after** the student has attended the mandatory orientation meeting will they be able to see all the host information (e.g. name, cell phone, email, etc.) about the host they have been matched to on their Status Portal (/student/[studentId]).
- seeAssignment – boolean field as to whether it is allowed for a student to see what host they have been assigned to by the algorithm. Default false.
 - Note that students see the host they have been assigned to on their Status Portal (/student/[studentId] route.)
- whenPublic – Date for when the event administrators have decided to allow a student to see what host they have been assigned to by the algorithm. Default NULL value. This timestamp tells students how much more time they have left (+2 days) until they *need* to confirm (accept/reject) the given host they have been assigned.

All these fields can be changed by the administrators of the program.

Note that for a given student in a given event, there is only at a maximum ONE record in MatchedStudent. If a student has decided to reject a given student they have matched to, their record in MatchedStudent table is deleted, and a new record in RejectedMatchedStudents is created.

RejectedMatchedStudents

This is to store a list of all students who have been matched and have decided to REJECT a given host they have been offered. The RejectedMatchedStudents table stores the student ID, event ID, and hostID of what the student choose to reject. This table is used to provide records on who to skip (i.e. students who have already rejected a given host) when trying to assign students to hosts via the algorithm.

AuxiliaryPerson

Useless table. Delete.

Summary

The table relationships are no joke, as you can see. Understanding every field of the table is crucial when making any API change. **Note that unique constraints for rows were not mentioned in this guide.** To further understand the Supabase database and limits, please refer to the “Services, Plans, and Limits” section.

User Authentication

Supabase Auth manages user authentication. It is quite self-explanatory and the Users can be viewed in the Authentication section of the Supabase dashboard for this project.

Next.js Backend API

Overview

This is an overview of the Backend API that is part of the Next.js project. There is no other API that is part of this Job Shadow platform that the Job Shadow developers have developed.

Explanation of Individual API Route

The guide below details each individual API route. To see what parameters each route takes, just search them up in the Next project. Note that most API routes are called via the useSWR hook in the Next frontend files. To call any route, use the path shown in the headers below prefixed by “/api”.

/

This route is just a test route to make sure the server is working.

/allDBEvents

This route provides all of the Job Shadow Events that are stored in the database.

/missingHosts

This route provides all of the Job Shadow hosts in the database that are missing from a certain event (hardcoded EVENT_ID on Line 8 in pages/api/missingHosts.ts).

/getAllHosts.tsx

This route provides info on all of the hosts in the database and the name of the recruiter that manages each host.

/users/getAll

Provides all users in the database (stored in the Users table.)

`/users/delete`

Entirely deletes a given user from the entire database (deletes them from all events, deletes any hosts they manage, deletes everything).

`/users/get`

Request that searches for a given user to get their information (e.g. what role is this user?)

`/student/`

Delete this route.

`/student/add`

Links a given student to a given Job Shadow event.

`/student/getStudent`

Provides information on a given student. If an event ID is given, this API route will give info on whether the student has submitted their host preferences.

`/student/email_redirect.ts`

Useless route but do not delete (maybe useful?)

`/student/status.ts`

This route provides information on the status (submitted preferences, unassigned, unconfirmed, confirmed, refused a host) of a student in ALL the events they are in. This route is used to inform the Status Portal that students see on their landing page after they login.

`/student/events/[eventId]/`

Send info on a given event and whether a student is approved for this event.

`/student/events/[eventId]/changeAttendedMtg`

Change info on a student as to whether they attended the mandatory orientation for a given event.

`/student/events/[eventId]/changeSeeAssignment`

Change a student's ability to see their host assignment for a given event.

`/student/events/[eventId]/decideHost`

Update a student's preference for their host assignment (accept/reject) for a given event. If rejected, the student's record in the MatchedStudents table is removed and a new record is made in the RejectedMatchedStudents table.

`/student/[studentId]/getHostsAndPrefs.ts`

Give the (sorted) host preferences a student has submitted for a given event.

`/recruiter/`

This is the route for **creating** a recruiter in the database. This will create records in the User & OpportunityProvider table.

`/recruiter/getRecruiter`

This route will give info on a recruiter.

`/recruiter/add`

Redundant route with `/api/recruiter/[recruiterId]/joinEvent.ts`. Please delete.

`/recruiter/add_host`

This route will enable a recruiter to create a new host (new record in Hosts table.)

/recruiter/events/approved

Useless route please delete.

/recruiter/events/

This route returns all the events that a given recruiter is part of.

/recruiter/[recruiterId]/joinEvent

Enables a recruiter to join an event (create record in RecruiterEventPair table.)

/recruiter/[recruiterId]/leaveEvent

Deletes a recruiter from a given event (delete record in RecruiterEventPair table.)

/recruiter/[recruiterId]/hosts/add

Assigns a host from a recruiter to a given event (create record in AssignedHosts.) If a recruiter is not already linked with this given event (i.e. there is a record for this link in RecruiterEventPair), the record will be created.

/recruiter/[recruiterId]/hosts/delete

Deletes a given host entirely from the database.

/recruiter/[recruiterId]/hosts/edit

Enables editing of a given host. If it is specified that it is a host point-of-contact editing the given host (NOT the recruiter), the `formSubmittedTimes` field in the Host Table will be updated.

/recruiter/[recruiterId]/hosts/get

This route gets all of the hosts that a recruiter is responsible for.

/recruiter/[recruiterId]/hosts/getEvents

This route returns all of the events that a recruiter has hosts added to.

/recruiter/[recruiterId]/hosts/remove

This route removes a given host from a given event.

/recruiter/[recruiterId]/hosts/removeSpec

This route completely deletes a given host from the database (delete entry in Host table.) **Note that there probably should be a change here that first deletes any records of the host in the AssignedHosts table.**

/events

Returns all the events that a given admin user is responsible for.

/events/students/[eventId]

Returns info on all of the students who are a part of a given event. This includes unapproved students.

/events/students/[eventId]/unverified

Returns a list of the students in the database whose emails are unverified. This route is used for a particular Data Export on the admin side. Note that it takes a long time to run.

/events/students/[eventId]/resetStudent

This route is meant to completely reset a given student in an event. This **does not** reset if they are approved/unapproved. This removes any host preferences the student has submitted, removes any records they have in MatchedStudents (if they have been matched) or RejectedMatchedStudents (if they have rejected a given student.)

/events/students/[eventId]/getHostPrefs

This route provides the given submitted host prefs **for all the students** that have submitted their host preferences for a given event. Note that the /student/[studentId]/getHostsAndPrefs is only for one student.

/events/hosts/[eventId]?approved=true/false

This route returns either ALL of the hosts assigned to a given event or only the approved hosts assigned to a given event depending on the query parameter “approved.”

/events/hosts/[eventId]/approved

Redundant route with above.

/events/hosts/[eventId]/sendEmail

This route sends an email to lhsjobshadow@gmail.com using the Resend API. This email is meant to inform a host point-of-contact of the students they will have to host on Job Shadow Day. **It is sent to lhsjobshadow@gmail.com as the LHS Job Shadow administrators will manually send this email to the hosts.**

/events/[eventId]/

This route provides information on a given event (returns the JobShadowEvent row.)

/events/[eventId]/confirmations

This route gives information on the status (unassigned, unconfirmed, refused, confirmed) of students who have submitted their host preferences for a given event. A big chunk of the “View Confirmations” page that the Administrators access is powered by this API route.

/events/[eventId]/getAllRecruitersAndHosts

This route returns all the recruiters in an event, all the hosts in an event, and the set of hosts that belong to each recruiter in this event.

/events/[eventId]/importantInfo

This route provides the list of all students which are assigned to hosts which either have important information (importantInfo field is non-null) or are on an atypical date. This is used to power the “Special Requirement Assignments” view page on the Admin side.

`/admin/`

This route is used to create a new administrator.

`/admin/add`

This route is for an admin to be able to create a new Job Shadow Event.

`/admin/[eventId]/assignHostPreferences`

This route enables an administrator to manually assign host preferences to students..

`/admin/[eventId]/edit`

This route enables administrators to edit info for a given Job Shadow event. Note that the typicalDate field cannot be edited.

`/admin/[eventId]/getAssignments`

This route provides data on the current Job Shadow assignments (info on students and info on the jobs they have been matched to) for a given event.

`/admin/[eventId]/transfer-host`

This route enables admin to transfer a given host from one recruiter to another recruiter.

`/admin/[eventId]/host-approval`

This route toggles an event administrator’s approval of a given host for their event.

/admin/[eventId]/student-approval

This route toggles the approval of a student for a certain event.

/admin/[eventId]/student-form

This route updates whether a given student (who has joined a given event) has submitted their paper form.

/admin/[eventId]/student-grade-update

This route enables admin to temporarily change a student's grade (so that they can be included in an earlier Job Shadow run.)

/admin/[eventId]/recruiter-approval

This route is completely useless.

/admin/[eventId]/recruiter

This route is completely useless.

/admin/[eventId]/manualAssign

This route enables the admin of an event to manually assign students to a given event.

/admin/[eventId]/runthealgorithm

This is the most important file in the entire platform. This route enables the administrators to automatically assign students (based on their submitted host preferences) to job shadow opportunities.

/admin/[eventId]/calibratecounts

This route should be executed after every Job Shadow assignment run. This route recalculates all of the counts of `slotsTaken` for each assignedHost in the Job Shadow Event based on the current Job Shadow Assignments in the MatchedStudent table.

/admin/[eventId]/calibratestudents

This route should be executed after every Job Shadow assignment run. This route sets matched=true for all StudentMembers who have been assigned to a job in this event.

/admin/[eventId]/unlink/host

This route unlinks a given host from a given event (deletes the record in AssignedHosts.)

/admin/[eventId]/unlink/student

This route unlinks a given student from a given event (deletes the record in StudentMember.)

What Needs to Be Done At Each Step of Job Shadow

This section details what to do during every step of Job Shadow, throughout the year. All the services and their plans/limits are given in the next section.

Before Job Shadow Student Registration

Make sure the platform is working and that recruiters can enter hosts correctly. **The only service that should be paid before Job Shadow Student Registration is the GoDaddy Domain. One week before student registration, make sure to enable Resend pro-tier and Supabase pro-version.**

During Student Registration

There is more user load during this week. Just monitor the platform, but there should not be any changes required at this stage.

In the case that a student loses their password:

1. Sign-in to the Admin Account and delete the user (the /admin/[slug]/delete route enables this)
2. Ask the Student to re-create their account

During The Week When Students Submit Their Preferences

User load significantly increases during this time. **Make sure to turn on Vercel pro-trial for the Vercel project at this time.**

After Job Shadow Day

Congratulations! **Turn off the Vercel pro-trial (if it has not expired yet); turn off the Supabase Pro Version subscription; turn off the Resend pro-tier subscription. The only service that should be paid at this time is the GoDaddy domain.**

Services, Credentials, Plans, and Limits

This section is meant to detail all of the different services, plans, and limits that this platform relies on (and their tiered level of pricing.) I also give the credentials to each different service below. They are listed below.

Supabase – PostgreSQL Database & User Authentication

The platform uses the Supabase platform to host its PostgreSQL database and user authentication. Essentially, the platform has its own supabase project.

Plans/Limits

The current supabase project used for the platform is on the **Pro Version** (\$25/month). This is only required when the Job Shadow program is running and there is heavy request load from students.

Credentials

You can login into Supabase with the lhsjobshadow@gmail.com email and password. Once logged in, all of the Supabase .env keys can be found there.

Vercel – Web Hosting

To host the website 24/7, we use Vercel. Vercel is a free service to use. However, during the days in which there is expected high student load on the platform, we utilize the free 14-day Pro trial for Vercel to increase the max bandwidth Vercel can handle.

Plans/Limits

Vercel is 24/7 a free platform to use. However, to increase the default bandwidth maximum to 250GB during the weeks where a lot of students are submitting preferences on the platform, please switch the Vercel project temporarily to a free 14-day Pro trial. Note that most bandwidth comes from the front page Job Shadow images and recruitersAndHosts API route.

Credentials

To sign into Vercel, talk with the Lead Developer who is responsible for having control of the GitHub repository for the Next project.

GoDaddy – Domain Management

Plans/Limits

The current plan on GoDaddy enables us to own the lhsjobshadow.org domain for the website. Using the domain IP, we can connect the Vercel site to this domain.

Credentials

You can sign into the GoDaddy account with the lhsjobshadow@gmail.com email account.

Resend – Email Server Management

There are two cases in which the platform will send emails to anyone:

1. User signs up and the platform needs to send email verification.
2. Admin clicks the button to send email to a Host point-of-contact. This email is sent prior to the Job Shadow Day and contains info on all the students who are assigned to this host.

In case 1, Supabase sends this email by default. In case 2, this email is written in the code and sent via the Resend API. **Note that the Supabase project is configured to use the Resend project as its email server.** This allows us to be able to send more emails (in a shorter space of time) than Supabase limits.

Plans/Limits

Resend.com pro-tier (\$20/month) allows for unlimited emails per day and 50,000 emails in a month. It is more than enough. Only enable pro-tier during the time when there is expected high student load on the site.

Credentials

You can login to the Resend.com web interface with the lhsjobshadow@gmail.com email. There, you can only create new resend API Keys. **The current Resend API Key is only available in the .env credentials file.**

Bugs/Future Implementations

Bugs

The bugs of this software are listed below:

- Bob can open any page on Alice's account without logging in as Alice.
- API routes are mostly not protected. A student can run an API without permissions. This can be solved through Middleware.

Things To Fix

1. Fix the above bugs. This is the most crucial.
2. Delete any code/API routes that reference the RecruiterMember Table
3. Delete the RecruiterMember & Auxiliary Person Table
4. Delete any frontend/backend routes that have been deemed useless

Distributing this platform to other organizations

If you are planning to distribute this Job Shadow software to other organizations, you have two options: (1) try to get all your organizations to create events on this platform or (2) clone this platform (get a new domain, a new Supabase Project, a new Vercel Project, etc.) for each organization. #2 is probably the easier option.

If you choose to try Route #1, **note the following violations that must be resolved:**

- By default, students who sign up to the platform are defaulted to joining a given hardcoded event.
- The admin of a given event have permission to delete ANY user in the database
- The /events/students/unverified API route searches for students who have unverified email. This search is applied to ALL students in the database (not just students for a particular event.)
- The /missinghosts page and /missinghosts API route both are dedicated to a hardcoded event.

.env Credentials & Service Passwords.

The following is the .env file to attach to the Next project:

DATABASE_URL=
PRISMA_DATABASE_URL=
NEXT_PUBLIC_ANON_KEY=
NEXT_PUBLIC_SUPABASE_URL=
RESEND_API_KEY=
SERVICE_ROLE=

Service Passwords

Supabase Password:

Supabase Database Password: