

1. Shaina: Introduce project and team

Hi everyone! My name is Shaina and my teammates are Leman, Matthew, and Maisa. We are here to present our web application: CSConnects-connecting students across computer science

2. Maisa: Product Definition

In order to connect the Computer Science community on our current virtual campus, CSConnects will solve students' problem of feeling disconnected from our peers, networking opportunities and club events and programming by giving them easy access to our clubs calendar as well as notifications when there are events that will interest them.

3. Leman: Product Definition cont.

We know our product will work when it:

- displays a full calendar of events for all CS clubs at Hunter College,
- When users can filter their event preferences
- notifies users about upcoming events.

In addition, CSConnects will display helpful resources for students such as contact information for club board members, tips to get more involved, links to group chats, discords, slack channels, and facebook groups of the clubs displayed and important links for students like the syllabus links and professor ratings.

4. Matt:

Demo

1. Order for demo:

- i. **Matt:** We will now showcase our app

1.

a. Homepage-Maisa:

For our home page, we wanted to briefly introduce students to the site, why they should join and the CS clubs hunter offers. Our home page features a selection of the clubs displayed on a card component. When a user hovers over one of these cards(**Hover over cards**), they may see a brief description of the club. If the user may be interested in that particular club, they can choose to read more(**middle click ACM**)and can easily find out more about that club and check out their sites and socials.

b. About page-Maisa

Now back on our site, the user may be interested to learn more about CSConnects and its goal. If we check out our about page (**click about page**) we see a section describing our reasons behind this project, "College isn't easy. There's already so much to navigate and being in CS at Hunter College means you'll be juggling all your work on top of all the opportunities you have to keep up on. Many of us have missed opportunities, events, and feel like everything is a mess. That's why we built CSConnects." We also see our mission, "CSConnects keeps Hunter students connected to each other, their clubs and their campus with ease."

c. Resources page-Maisa

We've heard a lot about what CSConnects aims to do, now let's see some examples of it. As the user, we will now head into our Resources page....[\(go to resources page\)](#) The resources page offers relevant links to CS Hunter College students such as Internships and Jobs, Mentorships and Fellowships, and resources specific to the Hunter College CS department. Once the user clicks on the plus button by any of these categories, such as for Internships and Jobs [\(expand internships and jobs\)](#), a curated list of job hunting websites appears. Once selected again, the display should condense. We can also look at resources relevant to CS Hunter Students once we select the category for the CS department [\(expand college resources\)](#). The student might note this description for CS Syllabi and select it [\(middle click Syllabi Directory\)](#) Now they have access to an up-to-date directory of CS syllabi.

d. Contact page-Maisa

And back to CSConnects..., let's visit our contact's page [\(open contact page\)](#) If someone has any questions, concerns or additional resources, they may seek out ways of contacting an administrator here.

Let's now check out what events are.

2. Calendar-Leman

- a. Here is the events page, which has a calendar that shows all the events from Hunter's computer science clubs. A user can go through the calendar and check if there were events in the past or future [\(click chevrons to go previous or forward in month\)](#). A user can also click on an event and that will show all the event information for that particular event. [\(click on Halloween party event\)](#) You can see for this event - the ____ club hosted it, its located ____, the start time and end time is ____ and the event type it is ____
- b. On the filter feature, if we click on Apps club, and career, we should only see that event
- c. I will now pass it onto my colleague, Matt, who will talk about the login/ sign up features and the user/admin roles.

3. Log in/Sign up - Matt

- a. I will now showcase the sign in slash login page for user and admins. Before I proceed, I would like to remind you of the purpose of an admin. There are hardcoded admins in our database. And only admins can add and delete events. Admins are select representatives for each club. Everyone else that signs up will have a default role of user. [\(click on sign in page\)](#) Here, we can see our sign up page. I will demonstrate the account sign up process. If the user tries to enter a phone number that is not valid phone number format, or they try registering a number that exists, they will receive an error. I will make the phone number 1234567899 and the password 123. After I sign up, the account will be stored into our mysql database hosted on heroku and the page will redirect to the login page. [\(click submit\)](#) Here, I will now login with the account I just created. [\(enter 123 credentials and sign in\)](#) After signing in, the user is redirected to the accounts page. In the accounts page, we can see your account information. This page

shows you your phone number that you will receive text notifications from, and additionally, there is a table that shows all events you want to be notified for. Currently, since this account was just created, there are no events that this user will be notified for.

- b. I will now showcase what the events page looks like for a user. [\(click events page\)](#)

As you can see, the events page still looks the same as when it was previously showcased by Leman. [\(click on a event to showcase it is the same\)](#). However there is an additional notify me button I will let my teammate Shaina talk about this feature.

Shaina: As a user or admin you have the ability to click “notify me” on each individual event to be added to our database. we will send you text messages daily, reminding you that you have an event the next day. If I am interested in a social event [\(Click club info session\)](#) then I can sign up for that [\(click notify me\)](#). I am also interested in a career panel with google. [\(matt do the clicking\)](#) I can click on it and then if we go to my accounts page now [\(click on my account tab\)](#) it will display the events i just signed up for in a table.

Now I will showcase what an admin experience will look like. [\(click sign in page\)](#) Here we used sessions to keep track of users, so I will now sign out of this account. [\(click logout\)](#).

Now that we have signed out, I will now login as an admin. [\(enter your phone number credentials\)](#). Now, we can see that our role has changed from user to an admin. As mentioned earlier, admins are selected representatives from each club. Now I will showcase the events page as an admin. [\(click events\)](#).

As you can see, there is now an additional interface. This interface is only visible to admins, and with this interface we can add events.

I will now add an example event for this week wednesday, the 16th.

For event name I will add Career Panel.

For the description I will type “Meet with industry professionals.”

Event description: Meet with professionals to expand your network.

Hunter location: Hunter West 203

The club hosting this event will be Hunter ACM.

This event type will be network.

The date of the event will be December 16, 2020.

Start time: 7:30pm

End time: 10:30pm

After filling in all the details, I will click submit.

As you can see, after inserting the events, it is now displayed on the calendar.

And if I click on this event, it will display more information about this event.

[\(click on event that was just added\)](#).

As an admin, there is also an additional button to delete this event.

I will now delete this event I just added.

5. **Shaina:** Now let's recap the contributions of each team member

Shaina:

- I Added the Twilio API feature and logic to send users SMS messages
- I also implemented a Cron scheduler to automatically send our users text messages
- I added some CSS additions and worked on the user experience and interface of the calendar page and accounts page
- I implemented the Delete event from calendar functionality for admins
- I also implemented functionality for a User to sign up for event notifications

Matt:

- Created login, signup functionality.
- Added login sessions and user roles.
- Added calendar admin panel
- Added backend API for filter feature
- Grabbed data from MySQL database to render accounts page data.
- Setup MySQL database on Heroku
- Contributed to MySQL database design
-

Leman:

- Researched calendar APIs - Google Calendar, React Scheduler, FullCalendar
- Set up events page, adding the calendar
- Contributed to database design and backend APIs
- Deployed project Heroku
- Helped with Git merge conflicts

Maisa:

- I created the mockup for the project and identified the reusable components we would need
- I set up the pages and created the components using ReactJS and SASS as a CSS preprocessor
- Used React Router for seamless user experience
- Worked on making a responsive, mobile friendly site

1. **Matt: Tools Used**

Here are some of the important tools and libraries that we used for our application. We used MySQL for our database to store information on users and events. Twilio API is used to send text messages and notifications.

Our frontend is React. In React, we used the FullCalendar library, which is used to render the calendar on the events page. Moment is used for getting date and time. Cron is used to schedule our Twilio text messages. Axios is used for API calls to our backend. For our backend, we used Express. And Formspree API is used for the contact page.

10. We accomplished all our goals that we set for ourselves in the beginning of the semester but we want to mention that if we had more time we would have liked to

First, improve the responsiveness of our application to be better compatible for mobile applications.

Second, implement functionality to allow our users to edit their notification preferences.

Third, Send a text message when a user signs up confirming that the phone number is correct with an added verification process.

Technical Presentation Begins-pause for questions-

Shaina: This Concludes our demo and project recap- Does anyone have any questions before we begin our technical presentation?

7. Twilio stuff-Shaina

- We decided to use Twilio's SMS notification service to send notifications to our users when their events are coming up. We wanted an SMS service because we thought it would be best for students to get text message notifications as opposed to emails that might get lost in their inbox.
- Additional services that Twilio provides include: email, telephone, video, chat, whatsapp and fax services. We only took advantage of their outgoing message features.
- We found Twilio really easy to use and the documentation available on their sight as well as on the internet really helpful when incorporating this feature into our app.
- We learned also that Twilio has amazing security features that notifies its users within seconds when their credentials are leaked to github because before our MVP Demo I accidentally pushed our credentials in a comment to github . It automatically gave us a new authentication token so the one that was made public was no longer valid.
- When you create an account you'll be given an authentication token and account id that you can find in the **console** which also displays your billing information and helpful resources.
- You can also use a trial phone number if you prefer not to pay which is what we ended up doing the entire time. With a Trial phone number, however, the numbers you want to text must be validated through the console on their web application prior to you sending them a message. If you bought a phone number this would not be the case but we did not find it necessary to buy one for the project for this class.

- I included some code from the backend here that is written in javascript to help you understand how to implement Twilio.
- First on the top left, we need a client instance to grab the account id and token from the environment file.
- On the right, The api greetings and messages were used to test that our account was set up and we used this in the mvp demo twilio form to test a text message. If you look at the messages api you'll see it creates a message and it requires 3 parameters: the from field which is our phone number specified in the environment file, the to field which is the user's phone number and the body which is the message itself. Then we send our message and catch the errors.
- The last picture on the bottom left sends a notification to an array made up of user phone numbers. These numbers were grabbed from the database as the session phone number of the user who opted in for notifications for a specific event.
- We used the notification service from twilio for this because we were notifying multiple users at once. This function first grabs the phone number of each index in the array and maps it to our variable "bindings". It formats it into a json object in the proper twilio format. Then calls the notifications service. A message is sent that reads: "Hello! You have an event coming up, Log into your csconnects account here: then the link to our site deployed on heroku"

next slide

8. Maisa: Twilio Cont.

- Limitations to the trial period:
 - Currently we are using a trial phone number and account which means we have to add the users we want to message to our validated contact list.
 - That's a limitation to the trial process BUT it was more than enough money to fund our entire project and we never upgraded.
- Show our current balance we have left + prices for text message
 - We can look at our own trial pricing info for example. If we take a look at these screenshots, Twilio shows us our Current Trial Balance. We currently have \$12.29 of the original \$15 we received.

next slide

9. Maisa: Twilio cont.-->

- As part of Twilio's plan, the price of each message sent or received being \$0.0075
- A phone number is \$1 a month
- Github student deal
 - Twilio does also offer a deal through Github's Student Developer Pack. Part of this deal includes \$50 in Twilio credit for students to use. We haven't had to claim this deal for scope of our project and testing.
- Overall, we are really happy with Twilio and found the documentation really helpful

11. Leman: Our second technical presentation would be about Calendar APIs. For our project we had to choose a calendar to display the club events. These were our options: Google Calendar API, DevExtreme Reactive and FullCalendar
12. Leman: We originally went with the Google Calendar API for our events. Many people use it and it has good documentation. However, we realized that the calendar we used for Google would be the same for all users unless we connected to their Google account. Also if a non-user filters the calendar then that would also filter another non-user calendar if that person refreshes the page. Also Google Calendar API doesn't have a dedicated React component which would make it easy for us since we were using React on the frontend
 - a. We decided to look for other alternatives.
13. Leman: We did some research on DevExtreme Reactive React Scheduler.
 - a. Not many people use this so finding answers would be hard
 - b. Documentation is also lackluster
 - c. We also were unable to get it to work
 - d. We ultimately decided to use Full Calendar instead

14. Full Calendar

Matt: For our technical presentation, we decided to showcase FullCalendar, which is a React Component that you can install. The left photo shows how easy it is to set up and start using FullCalendar, and the right shows the code we used to implement the calendar. The trickiest part of using FullCalendar was putting dynamic information into the events={} attribute.

We had to first fetch the data that we wanted from the database and put it into the events attribute. However, when we tried to put an array of events fetched from Mysql database into events attribute, fullcalendar was breaking and not rendering the data.

Fortunately, leman was able to figure out the issue and it turns out we had to format the data a certain way.

15.

Leman: In this slide, we can see the API call we used to grab events from our database and then formatted the data so that FullCalendar accepts it . You have to pass an array of objects into FullCalendar. FullCalendar also has a onClickEvent feature which we used to handle a user clicking on an event and the event information shows up. With Google Calendar this wouldn't have been that easy, because Google Calendar doesn't have a dedicated React Calendar component.

(Talk about how we handled clicking on a event, and the new information pops up)

16. Shaina: Thanks for listening! Does anyone have any questions?