

# Centigro

Sindhu Budhavarapu, Nick Dobslaw, Nick Juliano, Enrico Mazza, Patrick O'Donnell

<https://github.com/NickJ202/csc468-course-project>

# Tables of Contents

<b>Tables of Contents</b>	<b>1</b>
<b>Roles</b>	<b>2</b>
<b>Chapter 1: Project Vision</b>	<b>2</b>
Product Design	3
<b>Chapter 2: Implementation</b>	<b>4</b>
Features	3
Feature 1: Events	3
Functionalities	3
Feature 2: Users	4
Organizer	4
Event participants	5
Feature 3: UI	5
Mockups	6
<b>Chapter 3: Intermediate Milestones</b>	<b>7</b>
Accomplishments	7
Contributor Login & Account Creation	7
Event Creation	9
Django-Postgres API	10
Docker / Kubernetes	10
<b>Chapter 4: Project Progress</b>	<b>11</b>
Deployment	11
Application	11
<b>Chapter 5 Self Assessment</b>	<b>11</b>
<b>Author Resumes</b>	<b>12</b>

## Roles

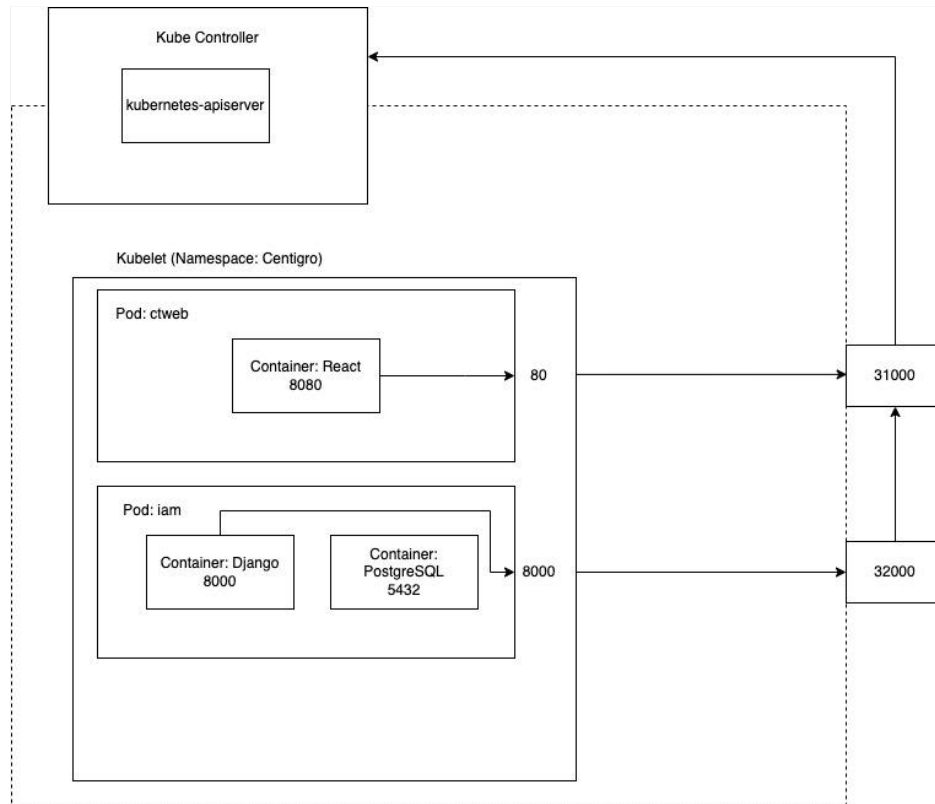
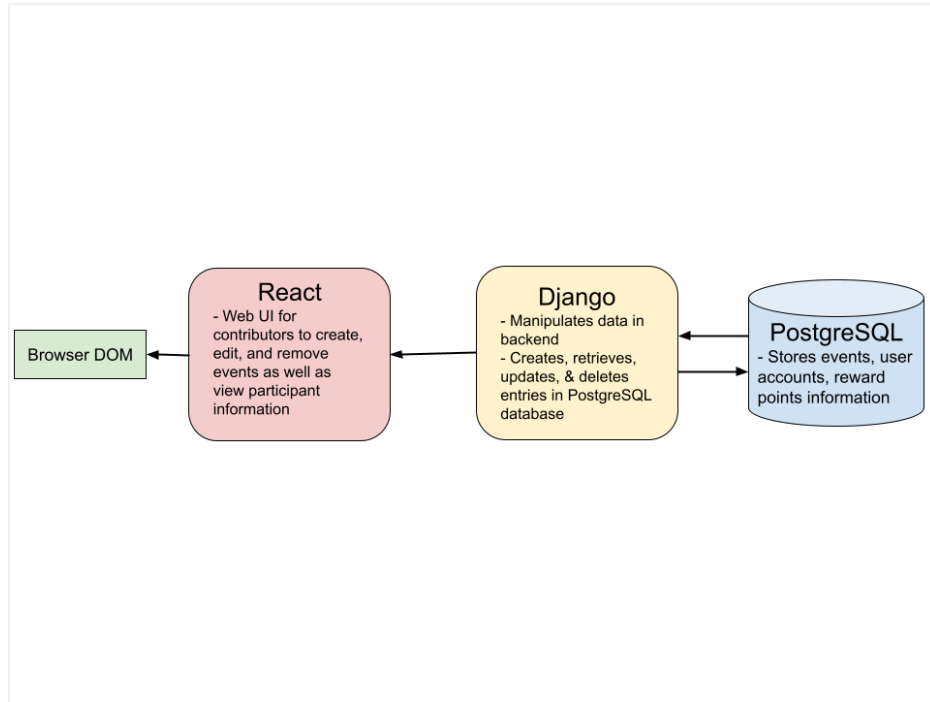
Sindhu Budhavarapu	Backend / API development
Nick Dobslaw	Frontend / Web UI development
Nick Juliano	API / Frontend / Mobile UI development
Enrico Mazza	Mobile UI Development
Patrick O'Donnell	Backend / API development

## Chapter 1: Project Vision

Centigro is a corporate social responsibility system that connects organizations with their local communities through incentivized environmental events. A contributor platform allows organizations to create and sponsor environmental tasks in their community. A participant platform allows end users to see and sign up for posted events. Upon completion of an event, users will receive points with the goal that they will be redeemable to vendors and charities in the future.

Core components include a web UI to handle the creation, deletion, and editing of events available to the end user, a mobile UI for end users to see and sign up to participate in created events, and an API to fetch / send all necessary event data from a PostgreSQL database. Below is a high-level diagram of the product design.

## Product Design



## Chapter 2: Implementation

### Features

#### Feature 1: Events

##### Functionalities

Event registration	Users can register for events as participants on the mobile and web platform. Events may require users to provide contact information or
Event Management	Event organizers can create, delete, and update their events as well as manage their participants. Event creation will allow organizers to specify details such as: <ul style="list-style-type: none"> <li>• Name</li> <li>• Description</li> <li>• Cover photo</li> <li>• Address</li> <li>• Start/end date</li> <li>• Tags</li> <li>• To-do list for attendees</li> </ul>
Event View	Participants will be able to view their registered events while organizers will be able to view their created events. A main page on both the web and mobile applications will display all events occurring during a set period of time.

#### Feature 2: Users

Centrigo users will be distinguishable by two roles: event organizer and event participant. These roles will allow for event creators to have full control of their events while participants will be able to share their necessary contact information with the organizer. These permissions and roles will be maintained at the database level, with roles being stored in a Django-PostgreSQL database based on a user's registration or creation of an event.

##### Organizer

Users that create events will be able to manage and make edits to their events. Permissions of the organizer include:

- Restricting participants to a specific organization
- Removing participants
- Updating event details, such as time, location, or description
- Viewing participants of an event, including contact information

## Event participants

Users that sign up for events will be able to manage their registration and view the event after confirmation. Permissions of the participant include:

- Registering for an event
- Unregistering for an event
- Updating contact information

## Feature 3: UI

For this project, since there are two parts to it (contributor and participant) we will be using web UI and mobile UI. Web UI would be for the contributor's side and mobile UI would be used from the participant's side. Both the web and mobile UI will be GUI based, and for mobile, there would be touchscreen GUI as well. Web UI, as well as mobile UI would have a login page which would allow the user to create an account and/or sign in/out.

For the web UI, the user will be allowed to create tasks in a separate window (first step) by entering basic information about the event like the title, description, location / time, and any images. There will be different buttons with the different options when creating a task. Within this window, for the time/date, there will be a calendar that will pop up.

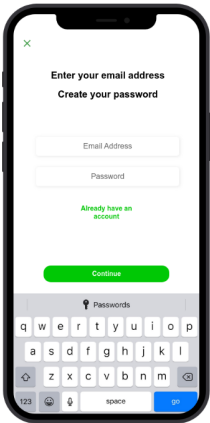
The next step would be on a new window which would consist of event information like to-do lists, and any extra information like what to bring, what to wear etc. This window would have different buttons to click on as well regarding everything that the event information consists of.

Finally, there will be another separate window that shows all the event information that was filled out by the contributor, which can also be seen by the participant.

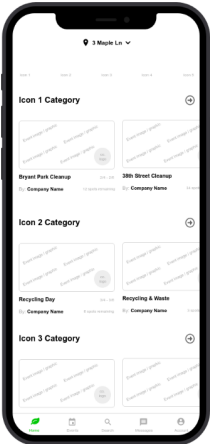
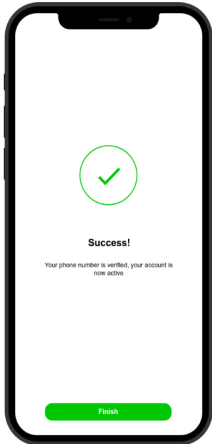
For the mobile UI, aside from the login page, there will be about four different windows. One window would be for browsing tasks, another window would be for signing up for tasks, another would be a window for physically completing the task and the last window would be for signing out. On all these windows, there will be different buttons for each task just like the web UI.

The web UI will be done with React JS, and the mobile UI will be done with React Native.

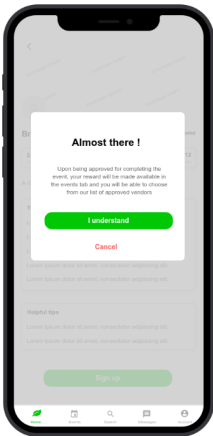
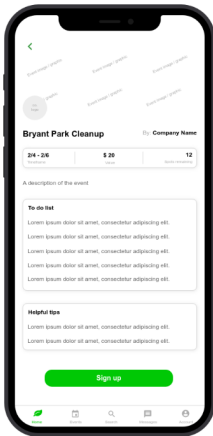
# Mockups



Initial mockups for signup and login page



Initial mock-ups for account confirmation and home page



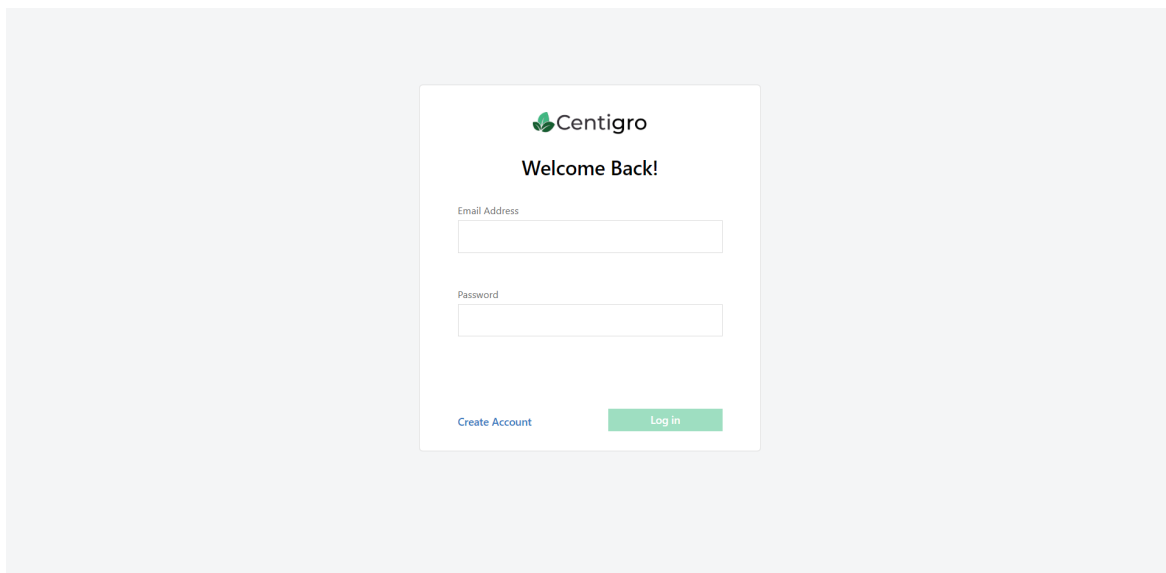
Initial mock-ups for event pages

## Chapter 3: Intermediate Milestones

### Accomplishments


#### Contributor Login & Account Creation

When an event creator launches Centigro, a login page is displayed where the user can either enter their credentials or create a new account. The account creation component consists of two pages. The first requires the organization's name and location, as well as a profile image upload to represent their profile. Then, they are prompted to enter their contact information and desired login credentials. Once the login form is submitted and verified, the home page (or Events page) is displayed to the contributor. If the user has previously logged in and created their own events, they will be displayed on this page in a continuous scrollable fashion. If the user has yet to create an event, a button is displayed under the Centigro logo that navigates to the event creation form.

The image shows a login page for Centigro. At the top, there is a Centigro logo consisting of a green leaf icon followed by the word "Centigro". Below the logo, the text "Welcome Back!" is displayed. Underneath, there are two input fields: "Email Address" and "Password". At the bottom of the form, there are two buttons: "Create Account" on the left and "Log in" on the right. The "Log in" button is highlighted with a green background.

**Login page that displays upon launch of Centigro**





### Organization

Name

Address


City

State

Postal

[Already have an account ? Log in here](#)

### Profile Image



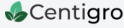
Upload

Edit

This image will be shown with events published by your organization.

Continue

**Contributor account creation form - Page 1 of 2**



### Contact

First Name

Last Name

Phone Number

### Login information

Email Address

Password

Confirm Password

Please enter your email and create a password, this will be the administrative user of the system.

[Back](#)

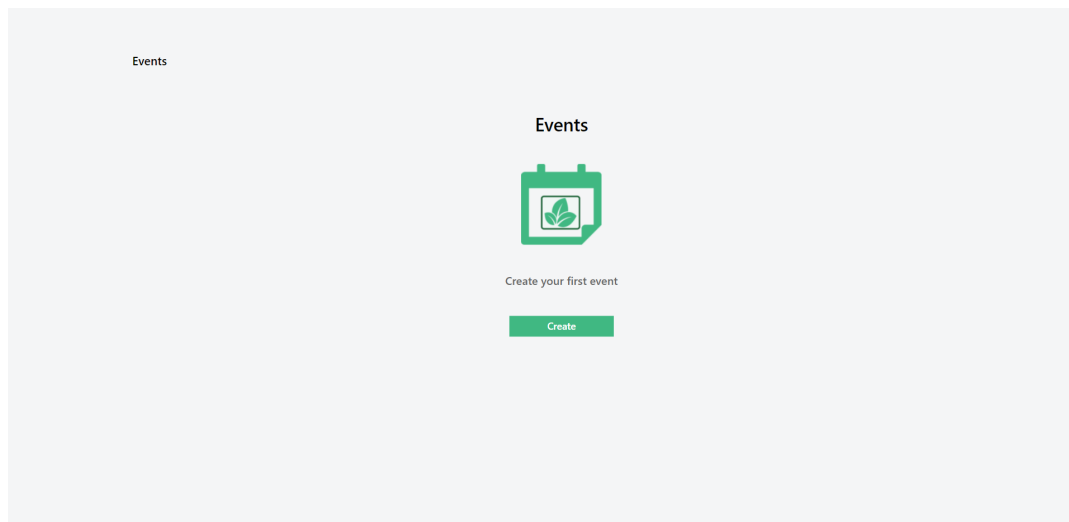
Submit

[Already have an account ? Log in here](#)

**Contributor account creation form - Page 2 of 2**

## Event Creation Page

The user-oriented process of creating an event for potential participants is carried out through a single page with multiple fields. These fields are divided into various sections: basic information, date & time, location, and participant information. The component utilizes a Redux Tree to observe field changes and update the event's object when user data is input, even upon the page's refresh. The creation page is styled consistently with the theme of Centigro's entire application. Contributors are presented with a summary of all information before verifying and submitting their event. Upon submission, the event object is passed to the participant portal for potential users to accept or decline attendance.



**Events homepage that displays upon successful login**

 The image shows a detailed event creation form. The form is divided into several sections: 'Basic Information', 'Time Frame', 'Participant Information', and 'Location'. 
 

- Basic Information:** Includes fields for 'Title of Event' and 'Description'.
- Time Frame:** Includes fields for 'Start Date' (MM / DD / YYYY), 'End Date' (MM / DD / YYYY), 'Start Time' (00 : 00 PM), and 'End Time' (00 : 00 PM).
- Participant Information:** Includes fields for 'Minimum Attendance' and 'Maximum Attendance'.
- Location:** Includes fields for 'Address', 'City', and 'State'.

 On the right side of the form, there is a calendar for May 2022. The calendar shows days of the week (MON, TUE, WED, THU, FRI, SAT, SUN) and dates. The date '6' is highlighted in green. At the bottom right of the form, there is a green button labeled 'Continue'.

**Event creation form**

## Django-Postgres API

We first created models for events, users, and organizations and successfully migrated these to a PostgreSQL database. The backend team developed tests using Django Test Cases for these models to ensure that all fields, especially ones depending on relations, were working properly. After the completion of the models, our team decided to use Django Rest Framework to facilitate the creation of an API. Django Rest Framework was the preferred choice as it allows for fast-paced and organized API development. The endpoints include GET/POST/UPDATE/DELETE requests for users, organizations, and events by ID or for all objects. User authentication was added through Django Rest Framework's token authentication. Upon the successful login of a user, the API returns a token that the React frontend can store and use to send an authenticated query to access any of the endpoints.

## Docker/Kubernetes

Both backend and frontend teams initially created separate Dockerfiles and docker-compose.yaml files to test that they were working separately. The only difficulty in the local containerization of our application was automatically migrating the PostgreSQL database, which we accomplished by using an endpoint, `entrypoint.sh`. This file handles the startup of the Postgres and the migration of `models.py` to the database. The merging of both docker-compose.yaml files into one file, `docker-compose.images.yaml` was fairly simple. After all containers were run correctly via a combined script, we began work on Kubernetes locally.

# Challenges

## Front-End Styling

Centigro's user interface follows a specific theme with an array of CSS styles for respective modules and HTML elements. I found it challenging at times to make sure all front-end developments maintained consistency to this theme with the vast amount of different components that are utilized.

## Front-End Organization

The GitHub repository for the Centigro app is organized in a specific way where all files are separated into assorted directories. For example, all components are stored within a 'components' folder that is broken down into three directories: `atoms`, `molecules`, and `organisms`. Each component is categorized into one of the folders based on its significance and has its own directory with various files: an index file with an export statement, a styles file with all element-specific designs, a types file that assigns all props to a uniform structure, as well as the main component file. Making sure that all components are organized efficiently within the repository structure and assigned the necessary files proved to be a constant challenge.

## Chapter 4: Project Progress

### Deployment

The majority of the progress made throughout this project has been from the deployment perspective. As of now, the project is containerized and deployed within Kubernetes, and each part of the application has a dedicated CDCI pipeline through Jenkins.

### Kubernetes

Each feature (web, api, and postgres) runs within a pod deployment of two replicas whereas Jenkins runs on one pod. All features are automatically deployed by Jenkins, which will be covered in the next section.

```
patodo@head:~$ kubectl get all -n centigro
```

NAME	READY	STATUS	RESTARTS	AGE
pod/api-ffdb6c8d7-jkp2v	1/1	Running	0	22h
pod/api-ffdb6c8d7-zvgjv	1/1	Running	0	22h
pod/jenkins-777f4d7994-rkfpf	1/1	Running	0	38h
pod/postgres-665bf5b8cd-24t9h	1/1	Running	1 (11h ago)	11h
pod/postgres-665bf5b8cd-dwnqq	1/1	Running	1 (11h ago)	11h
pod/web-6d89bd47fb-bbcxz	1/1	Running	0	38m
pod/web-6d89bd47fb-t8sw6	1/1	Running	0	38m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/api	ClusterIP	10.105.227.7	<none>	8000/TCP	38h
service/jenkins	NodePort	10.98.198.225	<none>	8080:31000/TCP	38h
service/jenkins-jnlp	NodePort	10.106.42.161	<none>	32000:32000/TCP	38h
service/postgres	ClusterIP	10.110.222.187	<none>	5432/TCP	38h
service/web	NodePort	10.104.93.15	<none>	80:30500/TCP	38h

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/api	2/2	2	2	38h
deployment.apps/jenkins	1/1	1	1	38h
deployment.apps/postgres	2/2	2	2	38h
deployment.apps/web	2/2	2	2	38h

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/api-76f5f998c6	0	0	0	38h
replicaset.apps/api-85d9b956fc	0	0	0	38h
replicaset.apps/api-ffdb6c8d7	2	2	2	22h
replicaset.apps/jenkins-777f4d7994	1	1	1	38h
replicaset.apps/postgres-665bf5b8cd	2	2	2	11h
replicaset.apps/postgres-7cf948d748	0	0	0	38h
replicaset.apps/web-54c9cd6698	0	0	0	7h46m
replicaset.apps/web-58c8c9d78c	0	0	0	8h
replicaset.apps/web-66d6459f7f	0	0	0	7h18m
replicaset.apps/web-6d89bd47fb	2	2	2	38m
replicaset.apps/web-75f97bb59f	0	0	0	20h
replicaset.apps/web-768f96554	0	0	0	7h24m
replicaset.apps/web-778bf78c5	0	0	0	8h
replicaset.apps/web-7c8cbf7d67	0	0	0	7h58m
replicaset.apps/web-7cc47ff799	0	0	0	7h14m
replicaset.apps/web-7fc7bbf598	0	0	0	9h
replicaset.apps/web-dc46bf4d	0	0	0	22h

*Display of all working deployments and services in Kubernetes*

The Kubernetes services turned out to run differently than our original plans. The only exposed pods are jenkins and web, with api and postgres running on ClusterIP. This ensures that all external communication is done within our React web application and all API communication with Django is contained within the cluster.

```
patodo@head:~$ kubectl get svc -n centigro
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
api	ClusterIP	10.105.227.7	<none>	8000/TCP	38h
jenkins	NodePort	10.98.198.225	<none>	8080:31000/TCP	38h
jenkins-jnlp	NodePort	10.106.42.161	<none>	32000:32000/TCP	38h
postgres	ClusterIP	10.110.222.187	<none>	5432/TCP	38h
web	NodePort	10.104.93.15	<none>	80:30500/TCP	38h

*Display of all services in Kubernetes, only three ports are externally accessible.*

```
patodo@head:~$ kubectl get pv,pvc -n centigro
```

NAME	CAPACITY	ACCESS MODES	RECLAIM POLICY	STATUS	CLAIM	STORAGECLASS	REASON	AGE
persistentvolume/postgres-pv	100M	RWX	Retain	Bound	centigro/postgres-pv-claim	local-storage		38h




  

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
persistentvolumeclaim/postgres-pv-claim	Bound	postgres-pv	100M	RWX	local-storage	38h

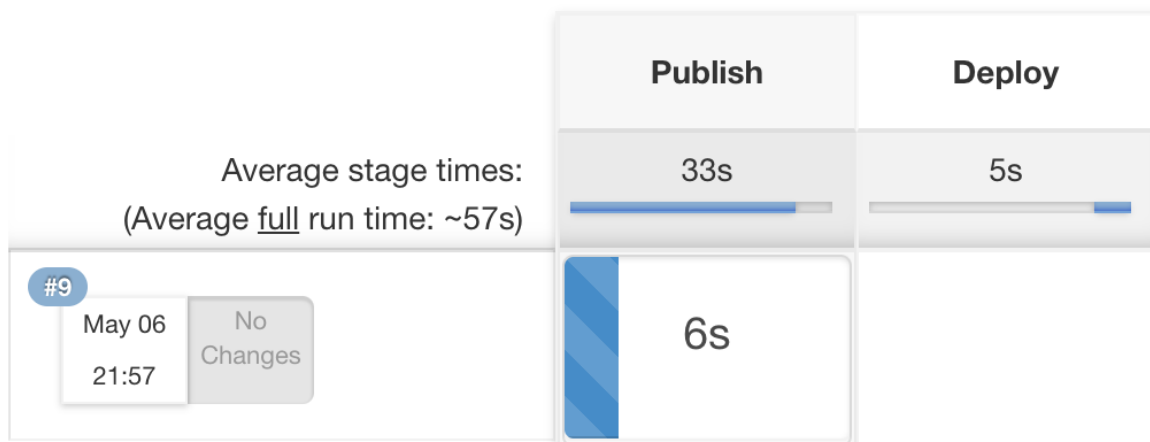
*Display of Postgres persistent volume and persistent volume claim*

## Jenkins

Jenkins is deployed as three separate pipelines pulling from different branches. The GitHub webhook is not configured, but each pipeline can be run manually to pull any changes from Github and deploy onto Cloudlab.

S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓		api	22 hr #8	23 hr #7	50 sec	▶
✓		postgres	11 hr #8	12 hr #7	57 sec	▶
✓		web	50 min #17	8 hr 24 min #10	1 min 36 sec	▶

*One pipeline in Jenkins for each feature*



*Manual build triggered for API pipeline in Jenkins*

```
patodo@head:~$ kubectl get pods -n centigro
```

NAME	READY	STATUS	RESTARTS	AGE
api-696dbb5c6-2vd2b	1/1	Running	0	39s
api-696dbb5c6-fbtft	1/1	Running	0	44s
api-ffdb6c8d7-zvgjv	1/1	Terminating	0	22h

*Kubernetes terminating old pods after new Jenkins build*

## Integration

As we have encountered multiple issues with integration between the React frontend and Django backend, at the time of submitting the project these two features are not fully integrated. The Web pod can contact the Django API from its service name, but we have not been able to correctly configure our React code to reach this endpoint.

```
patodo@head:~$ kubectl exec -it -n centigro web-6d89bd47fb-bbcxz -- /bin/sh
/ # curl http://api:8000/users/
{"detail":"Authentication credentials were not provided."}/ #
```

*Successful curl made from Web pod to Api pod in Kubernetes*

## Application

The main progress checkpoints of this project so far from an application perspective include authentication / authorization within the app, and the creation of events, which is a core component of the system. We have modeled the database to separate organizations and allow them to only see information related to their own organization. Each event contains a foreign key to an organization, so that they can be fetched only based on organization, as well as fetched without parameters to return any event in the table. There was also a lot of progress made in saving each event. Since there is a large amount of information tied to each event, it was difficult to maintain each piece of internal state within the frontend that would eventually be sent in an HTTP request to be saved through the API. By learning how to control the application's state in a more concise way using Redux, we were

able to find a way to maintain the large amount of information that must be stored with each event. The current state of the product is as follows, Organizations are able to sign up to be a part of the system, and they are able to save events to the database.

## Chapter 5: Self Assessment

This project has met the final deliverable by using an alternative design to the given project, as well as implementing full stack deployment including CDCI services. This project includes three core components. These components include a database, API, and user interface. The inclusion of the CDCI services can be demonstrated with a live update of the user interface, with the inclusion of a Docker image hub as part of the infrastructure. There has also been a conversion in infrastructure from using docker-compose to kubernetes deployment. Overall, this project has met the final deliverable by meeting each technical requirement regarding the deployment process, as well as the replacement of three or more services, where in this case there was an entire new design of an application created.

# Author Resumes

**Sindhu Budhavarapu**

[sbvarapu10@gmail.com](mailto:sbvarapu10@gmail.com) | 215-460-2013 | [www.linkedin.com/in/sindhu-budh](http://www.linkedin.com/in/sindhu-budh)

**Summary** Highly focused fourth-year undergraduate student excited to apply theoretical computer science concepts to industry. Dedicated to excelling in the field of computer science with special focus in software engineering and data analysis.

## Education

Bachelor of Science, Computer Science – West Chester University of Pennsylvania | May 2022  
*Independent Study | Artificial Intelligence* - worked with WEKA software to test samples of machine learning and data mining

## Relevant Coursework

- Data Structures & Algorithms, Programming Language Concepts / Paradigms, Software Engineering, User Interfaces, Big Data Engineering

## Skills

**Software:** Java, OCaml, Python, C, WEKA

**MS Office Tools:** MS-Word, MS-Excel, Power Point

**Spoken Languages:** French (basic conversational), Telugu (fluent)

## Professional Experience

**Robotics Trainer** - Wize Computing Academy | September 2021 - Current

- Teach motor and gear concepts to demonstrate movement of robots to elementary school students
- Direct students through basic coding skills such as loops and broadcasting to program robots built by students
- Prepare students to apply computing concepts to address real-world problems

**IT Intern** – Tela Bio | May 2021 - August 2021

- Worked with excel to sort out and analyze sales information using Macros
- Wrote simple queries in SQL Sever
- Developed Search Engine Optimization (SEO) keywords for company website
- Helped trouble shoot machines

**Resident Assistant** – West Chester University | August 2019 – May 2020

- Managed time well through balancing full course load with a position that requires a diverse skill set and high levels of attentiveness
- Resolved student and residential problems during on call duties and acting as an ever-present support system for residents
- Ensured residents adhered to rules and regulations while creating peaceful and accommodating environment to help build a stronger community

**Math and Reading Tutor** – Kumon Math and Reading Center | July 2015 – December 2017

- Tutored 15 students from the ages of 5-13 to improve their performance and understanding in math and reading with targeted instruction
- Organized and graded coursework and exams all the while identifying weak points for students to enhance understanding and convey results to guardians

## Leadership Experience

**Treasurer**, South Asian Student Association – WCU | January 2019 – May 2021

- Support for people of South Asian community, especially for international students who depend on guidance from familiar culture
- Hosted weekly meetings to get together with South Asian food and activities while hosting annual Diwali nights and annual Holi days

**Music Teacher** | June 2016 – Current

- Teaching Indian classical music to 4 students ranging from the ages of 8 – 13

**Senator of Student Leadership**, Student Government Association – WCU | August 2020 – January 2021

- Assisted in coordinating over 300 campus organizations to help students develop teamwork and leadership skills
- Led community service events outside of campus, like food drives during the pandemic



**Current Address**  
433 S. High Street  
West Chester, PA 19383

**ENRICO F. MAZZA**  
(610) 209-9552 | ricomazz100@gmail.com

**Permanent Address**  
1612 Herron Lane  
West Chester, PA 19380

## EDUCATION

**West Chester University | College of Science and Mathematics**  
*Bachelor of Science in Computer Science*  
*Certificate in Computer Security*

**West Chester, PA**  
May 2022  
Major GPA: 3.14

## WORK EXPERIENCE

### Treehouse World

*IT Technician Intern*

**West Chester, PA**

*July 2021 – Present*

- Independent developer working with android studio to create an application to move the daily park activity inspection forms from paper to fully digitalized to maximize efficiency and automation
- Provide routine maintenance for all technological issues with POS systems, printers, and internet park-wide

*Supervisor*

*May 2021 - Present*

- Optimize client experience by supervising 25 employees and managing all day-to-day business operations
- Maximize company efficiency by ensuring all opening, operating, and closing procedures are executed
- Resolve guest and staff issues to ensure safe and enjoyable experiences for all employees and customers
- Tailor product offerings to entertain over 600 customers daily and managed events up to 200 people
- Adapt to changing roles and responsibilities as needed while the company progressed and grew

*Adventure Staff*

*April 2020 – May 2021*

- Operated six different adventure rides throughout the park daily while following all safety procedures
- Assured customer safety on every ride by following safety procedures as outlined in each ride manual
- Built a positive and safe environment and created great experiences for all customers in the park

## CAMPUS INVOLVEMENT

### Delta Chi Fraternity

*Director of Social Activities*

**West Chester, PA**

*December 2021-Present*

- Plan and organize mixers and other similar events with other Greek organizations and clubs on campus
- Contact and work with different venues to set up and organize the Spring 2022 Delta Chi formal

*Vice President*

*September 2021 – December 2021*

- Executed chapter management procedures and planned and oversaw executive committee meetings
- Held chairmen on the executive committee accountable to achieve their tangible plans and goals
- Submitted weekly officer and committee reports at executive committee meetings

*Brotherhood Chair*

*September 2021 - December 2021*

- Responsible for organizing all fraternity events for brothers and any accompanying guests
- Strengthen the bond between the brotherhood by planning various events throughout the semester

*Fundraising & Scholarship Chair*

*January 2021 – May 2021*

- Promoted fundraising for the Jimmy V Foundation other philanthropies while raising over \$4,000
- Tracked and logged all brothers' GPA status, ensuring each brother maintains a cumulative of 2.7 or higher
- Created study groups to allow brothers to help brothers keep good academic standing during the semester

## HONORS & AWARDS

Dean's List: Spring 2020 & Spring 2021

Eagle Scout: Highest rank in Scouts BSA (Boy Scouts of America) (May 2018)

Test Out Network Pro Certification (May 2017) Nationally Ranked top 30% test score

Test Out PC Pro Certification (May 2016)

## RELEVANT COURSEWORK

Data Structures & Algorithms, Computer Security & Ethics, Software Engineering, Network & System Administration, Digital Image Processing

## SKILLS

Java, Windows 10, macOS, Cat5e Cable Configuration, PC troubleshooting, Wireless Networking, IP Configuration, Network Management & Security, Cisco Routing & Switching

# Patrick O'Donnell

Software Engineer

West Chester, PA  
443-621-0171  
odonnellp46@gmail.com  
github.com/manlalaro1  
linkedin.com/in/pvtrick-odonnell

## Work Experience

### Major League Hacking & Facebook, Remote — Production Engineering Fellow

JUNE 2021 - AUG 2021

- Designed and deployed HTTPS-enabled portfolio website on EC2 instance using Python, Flask, Nginx/Gunicorn, SQL, and AWS.
- Rolled out containers for Flask in Docker using Github Actions and cAdvisor.
- Developed a production-ready IOS event scheduler app with monitoring, testing, and CI/CD using Flutter, Django, and GraphQL.

### Archanics, Remote — Software Engineer Intern

JUNE 2020 - JULY 2020

- Built Django-Swift API to transfer geolocation data from Mac client to Leaflet-powered map while working with tools such as Postgis and GeoDjango.
- Led multiple technical designs on a 5-person team for various infrastructure, web, and mobile applications deployed using AWS Lightsails.
- Arranged reports and other documentation to assist development team members.

### HackWCU, West Chester, PA — Hackathon Organizer

DEC 2019 - MAY 2020

- Led sponsorship team, raising \$19,000 in fundraising efforts.
- Coordinated with marketing team, ultimately gathering over 200 attendees.
- Pitched and maintained communication with 26 sponsors via online and on-site meetings with company representatives.

## Projects

### Project GoGo — React / GraphQL / FaunaDB

Implemented unofficial database and website for board game Power Rangers: Heroes of the Grid using React, FaunaDB, and GraphQL.

### Butterfly — Flutter / Django / GraphQL

Designed backend for IOS event scheduler app using Django-GraphQL and SQLite by implementing GraphQL queries, mutations, and testing

### Security Dashboard — Spark / Python / InfluxDB / Grafana

Created a dashboard to stream log-in attempts from honeypot servers to a Grafana dashboard under the supervision of professor.

## Languages & Skills

**Languages** // Python, Java, React, GraphQL, SQL, PostgreSQL

**Frameworks** // Git, Spark, AWS, Django, Flutter, FaunaDB, InfluxDB, Grafana

## Education

### West Chester University, West Chester, PA — B.S. in Computer Science

AUG 2019 - DEC 2022

GPA: 3.802

**Activities** // International Computer Science Honor Society, Computer Science Club

## Nick Juliano

nickjuliano20@gmail.com (610) 717-7921 Garnet Valley, PA [LinkedIn](#)

### WORK EXPERIENCE

---

#### Frogfish Solutions – Software Developer

June 2019 – Present

Chadds Ford, PA

- Worked with a team to develop and maintain multiple e-commerce applications based on open-source technology.
- Developed an e-commerce returns application for customers to return or exchange previously ordered products.
- Developed an on-demand UPC ticket printing program where users can enter either SKU information or the UPC to generate and print the bar code label.
- Developed an external leasing application feature which allowed users more options on calculating quotes for various equipment products.
- Currently developing a dynamic data storage API as well as a cloud manager which will serve as the backend for different applications in the future.
- Developed ERP application feature to organize shipping codes sent from multiple e-commerce frameworks.
- Automated the deployment of multiple types of web applications which includes the handling of infrastructure, web server configuration, and deployment / re-deployment of code.

#### Zendi – Software Developer

January 2020 – June 2020

Glen Mills, PA

- Developed a mobile application trivia game feature which allowed users to receive points that were added to their points balance within the application.

### EDUCATION

---

#### West Chester University – B.S., Computer Science, Certificate in Computer Security

August 2019 – May 2022

West Chester, PA

GPA: 3.967

### SKILLS

---

Languages: Python, Java, JavaScript, TypeScript; Frameworks: Django, Apache Struts, React Native; Databases: SQL (PostgreSQL, MySQL), NoSQL (MongoDB); Libraries: React; Tools: Git, Unix

# Nick Dobslaw

## COMPUTER SCIENCE INTERN

Elizabethtown, PA 17022

(717) 777-1247

ND913745@wcupa.edu

### OBJECTIVE

Student motivated to acquire additional skills and experience with computer development. Highly flexible and efficient with array of computer topics. Currently studying concepts of networking, security, and code optimization. Driven to accumulate first-hand knowledge in a professional setting.

### EDUCATION —

West Chester University of  
Pennsylvania - West Chester,  
PA

Bachelor of Science:  
Computer Science  
Expected in 05/2022

Master of Science:  
Computer Science  
Expected in 05/2023

### KEY SKILLS —

Java, Python, C, Javascript, HTML  
Languages

Experienced with Visual Studio Code  
and GitHub

Command-line interface in Windows  
& Linux OS

Novice in Excel

Strong Quantitative & Problem-  
Solving Skills

Adaptability and Desire to Learn

Strong Work Ethic & Practicality

### WORK HISTORY

06/2021 - Current

Junior Web Developer • Quantum Strategies • Glen Mills,  
PA

- Style ticketing system front end and assure for an efficient user interface
- Develop web application to handle database connection
- Manage cleanliness and organization of repository code
- Maintain knowledge regarding Javascript, HTML, & SQL languages

### LEADERSHIP

Delta Chi Fraternity – West Chester University Chapter

Head of Alumni Relations – Executive Board Member

- Primary contact between alumni and chapter
- Manage communication about chapter efforts and events
- Coordinate all alumni events
- Work closely with president and other executive board members to manage internal business

Recruitment Committee Member

- Market our chapter through on-campus and social media efforts
- Plan and schedule all events during rush week
- Manage reservations, budgeting, and logistics
- Build network of potential new members and other Greek life chapters on campus