Final Report
Teacher Supply Donation Web Application
Project Archive - Final

CS467 Online Capstone Project
Fall 2022

Team:
Alice Fisher
Joel Swenddal
Sean Park

Table of Contents

Introduction

In the US, K-12 education system school budgets are often insufficient to cover classroom supplies resulting in teachers purchasing supplies themselves or requesting donations from students and their families. While there are online solutions available such as DonorsChoose or AdoptAClassroom, the application and setup process is tedious, lengthy, and usually focuses on monetary funding campaigns rather than actual items. That's why we developed Teacher Supply Donation (TSD), a full-stack web application that provides teachers with a much more efficient and streamlined process for creating donation campaigns for classroom supplies rather than money. With Teacher Supply Donation, teachers can create a list of supplies needed for their class and publish a personalized donation page. Students can register to donate items from that list and update their donations at a later time if they choose to do so.

URLs

Teacher Supply Donation Web App: https://tsdonation.live
Back-end Web API: https://donation-web-api.onrender.com

User Perspective

There are two primary users for the Teacher Supply Donation Web App:

Teachers
Teachers are the primary user of the application having the ability to create a user account with an email and password or by connecting their Google Gmail account. Once an account has been created, teachers can manage their supplies list, donors/students, and donations with an admin dashboard. The primary task that teachers will perform is creating a list of supplies needed for the classroom including the name of the item and the quantity needed. The list can be published which will then generate a unique URL for the teacher's donation page. This URL can then be shared with their students so that the students can sign up to donate items from the list.
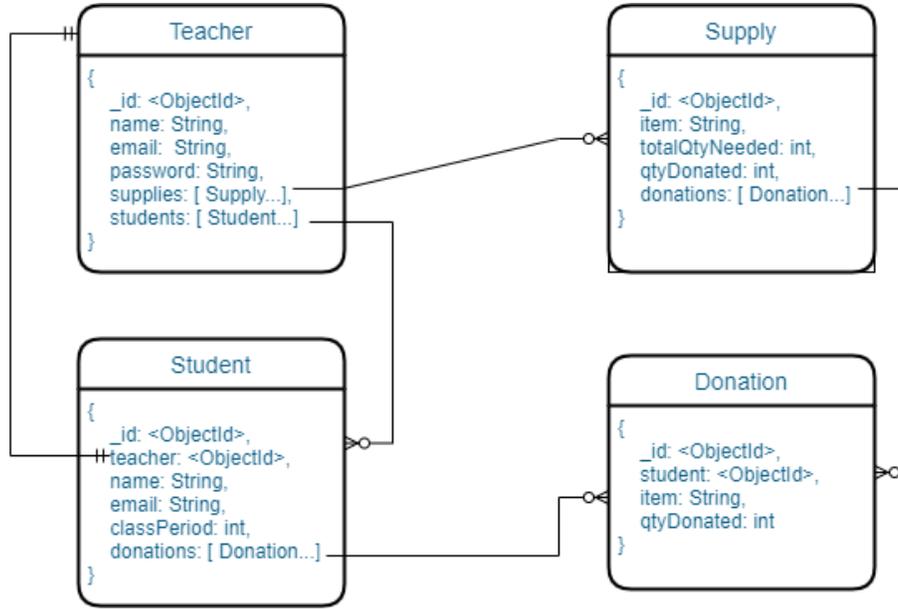
Students
Students are the secondary user for the application. Rather than creating an account with a password, students register as a donor by providing their name and email address when making a first-time donation. Once they have registered, a unique donation code will be provided which the students can use to update their donations and update their student profile. Every donation event will provide them with an email confirmation listing the name and quantity of items they have donated.

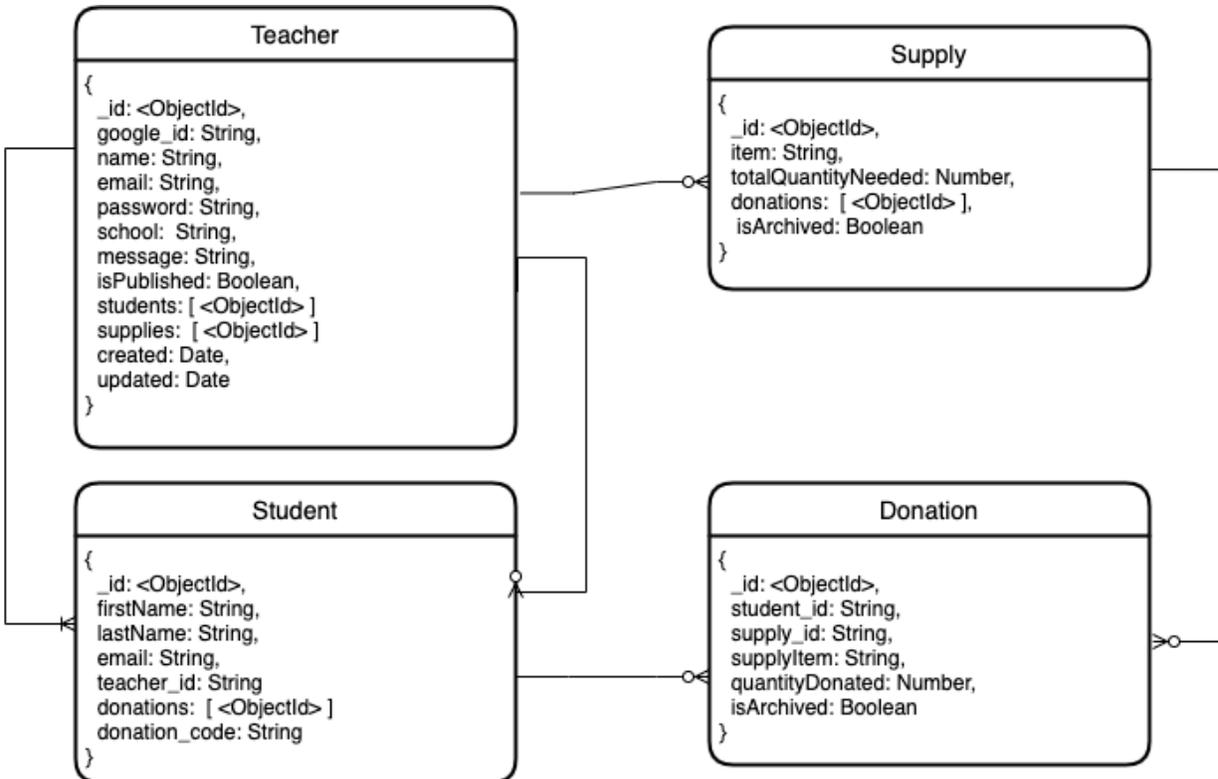# Comparison of Development Efforts and Project Plan

There were some changes and deviations from the project plan during the implementation of the web application, especially when it came to the Express back-end web service. The most

significant changes occurred with the MongoDB database design and schema as reflected in the diagrams below:

**Project Plan MongoDB Data Model Design/Schema**

Teacher
```
{
  _id: <ObjectId>,
  name: String,
  email:  String,
  password: String,
  supplies: [ Supply...],
  students: [ Student...]
}
```

Supply
```
{
  _id: <ObjectId>,
  item: String,
  totalQtyNeeded: int,
  qtyDonated: int,
  donations: [ Donation...]
}
```

Student
```
{
  _id: <ObjectId>,
  teacher: <ObjectId>,
  name: String,
  email: String,
  classPeriod: int,
  donations: [ Donation...]
}
```

Donation
```
{
  _id: <ObjectId>,
  student: <ObjectId>,
  item: String,
  qtyDonated: int
}
```

Actual Implemented MongoDB Data Model Design/Schema

Teacher
```
{
  _id: <ObjectId>,
  google_id: String,
  name: String,
  email: String,
  password: String,
  school:  String,
  message: String,
  isPublished: Boolean,
  students: [ <ObjectId> ]
  supplies:  [ <ObjectId> ]
  created: Date,
  updated: Date
}
```

Supply
```
{
  _id: <ObjectId>,
  item: String,
  totalQuantityNeeded: Number,
  donations:  [ <ObjectId> ],
  isArchived: Boolean
}
```

Student
```
{
  _id: <ObjectId>,
  firstName: String,
  lastName: String,
  email: String,
  teacher_id: String
  donations:  [ <ObjectId> ]
  donation_code: String
}
```

Donation
```
{
  _id: <ObjectId>,
  student_id: String,
  supply_id: String,
  supplyItem: String,
  quantityDonated: Number,
  isArchived: Boolean
}
```

We wanted to gather and display more information about the Teacher so several fields such as "school" and "message" were added to personalize each teacher's donation page. A "donation_code" field was added to the Student model which is the student's last name and the last 4 characters of their database object. Because the MongoDB object id is 24 characters long, it would be difficult for students to remember their id when using it in the donation page to update their donations. From the student perspective, having their last name plus 4 characters as their donation code was a better solution since it is much easier to remember and input into a form. We also added a feature to email a student their donation code by filling out a form if they forgot what their donation code was. In the initial project plan, we did not account for the scenario where a teacher would be intaking new students for a new school year and thus did not have a reset feature designed. To accommodate this, an archive feature was implemented allowing the teacher to archive all the supplies and donations, giving them a blank slate to start a new list of supplies. The archived data would still be available for the teacher to view by navigating to the Archive dashboard.

When planning for the front-end design of the application, we only had a mock-up created for the Teacher admin dashboard. Any add-ons, changes, and deviations happened as we were developing and were a result of the needs of the application.

Project Plan Admin Dashboard UI Mock-Up
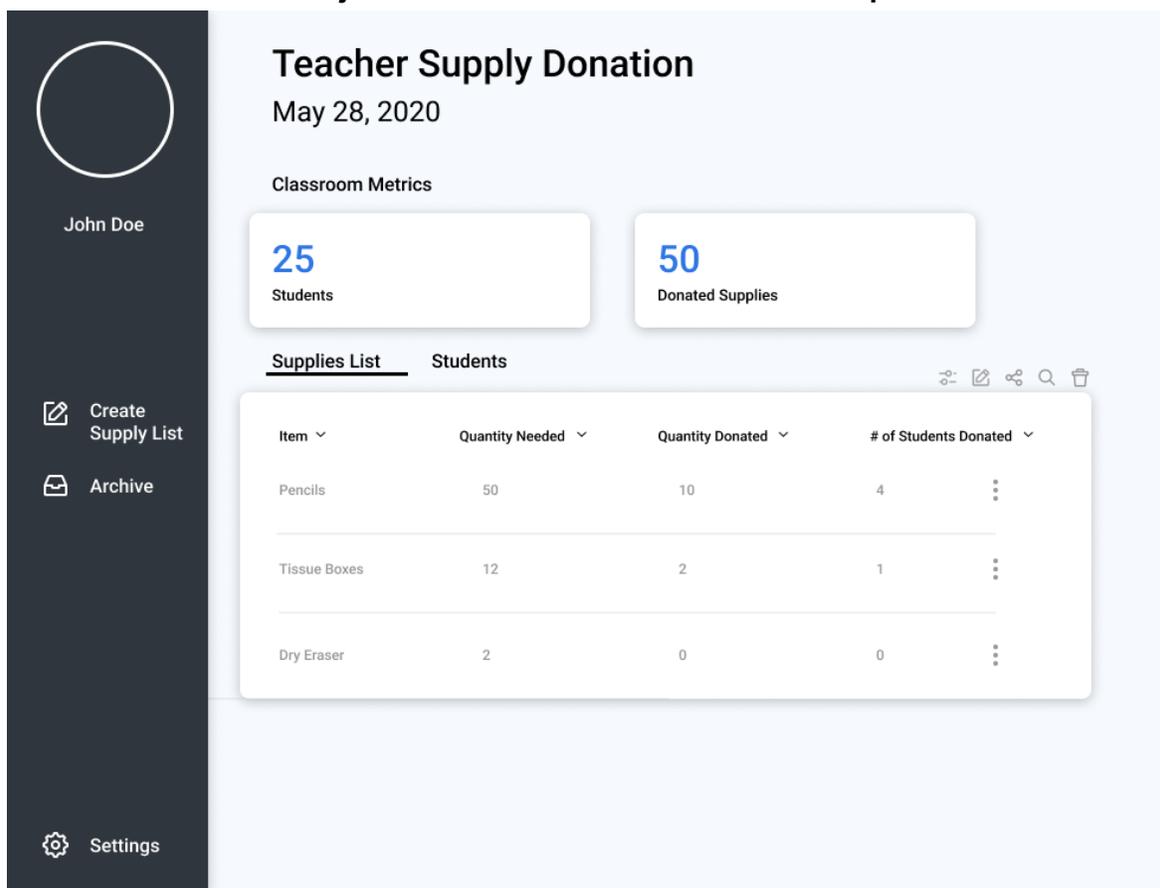
Actual Admin Dashboard UI Design
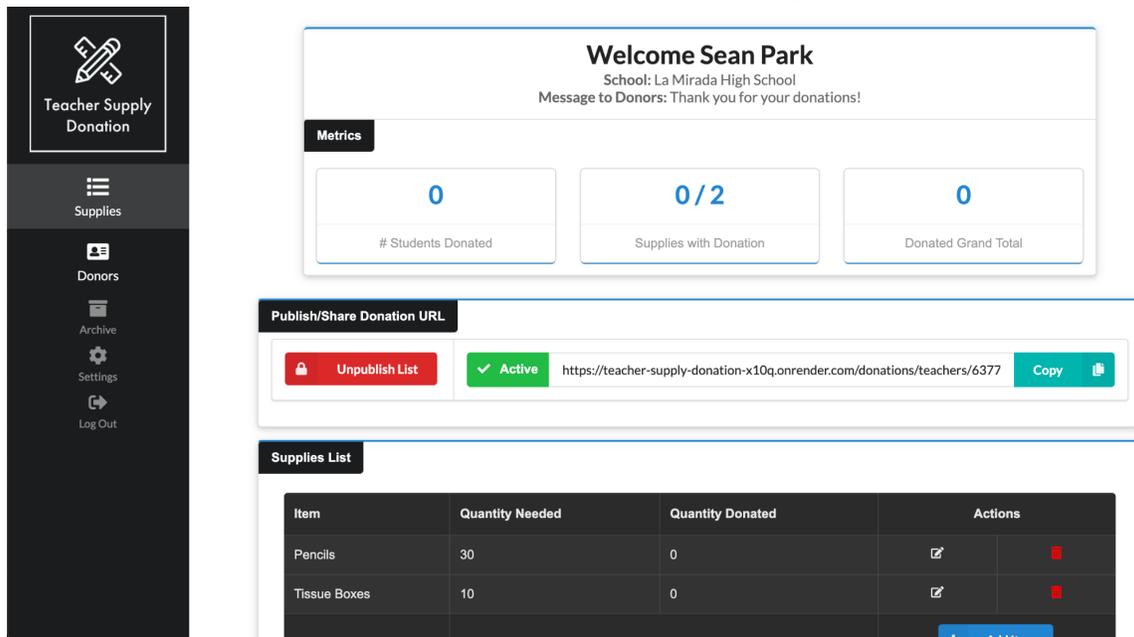
Overall, our development efforts followed close to our project plan with changes and deviations occurring as a result of discovering opportunities to enhance the application, not as a result of obstacles.

Software Tools & Technologies

Teacher Supply Donation was developed with the MERN stack (MongoDB, Express, React, and Node).

Front-End
- React (JSX/JavaScript)
- Semantic UI (UI Kit/CSS Framework) for functional components and style consistency
- @react-oauth/google for Google login authentication
- react-csv-downloader to generate a downloadable .csv file
- Name.com for customized domain

Back-End
- Node.js runtime with Express.js framework (JavaScript)
- Passport.js middleware for authentication and authorization
  - Local Strategy (Email and Password)
  - Google Strategy
  - JSON Web Token (JWT) Strategy - JWT tokens expire every 2 hours so that user sessions are not persisted indefinitely for security reasons
- Mongoose to connect to MongoDB database
- Chai, Jest, Mocha, and Supertest for testing
- Nodemailer for sending emails

Database
- MongoDB Atlas (NoSQL Cloud Database)

Hosting
- Render.com for development and production deployment.
  - CI/CD pipeline connected to GitHub for auto-deployments

Other Tools & Technologies
- GitHub for version control
- Postman for API testing
- Jira for project management
- Figma for project design plans
- Microsoft Teams and Zoom for team communication

Conclusion and the Future

The past few weeks were filled with a tremendous amount of learning and hard work that reflect our passion for the project. Our team is extremely proud of the outcome of Teacher Supply Donation and confident that it will become a valuable resource for K-12 grade educators. The goal of this project is to solve a problem teachers face when it comes to acquiring supplies needed to facilitate student learning in a convenient and efficient manner. We believe that we achieved that goal with Teacher Supply Donation and hope that teachers and students from all over will enjoy the experience of using the application. The development efforts resulted in a successful launch of the web application without any compromises to what we initially planned for. Rather, we were able to add features that further enhance the user experience for both the teachers and students. Using MongoDB, a NoSql database allows us to scale up the data we want to capture and alter the schema in the future if needed due to its flexibility. We also re-launched Teacher Supply Donation with a custom domain (tsdonation.live) to further legitimize the application without the hosting platforms branding the URL.

We plan to continue to improve the Teacher Supply Donation project. Currently email notifications are being sent to students as plain text. We would like to add the ability for teachers to share their public donation URL more easily via email or social media and create HTML emails with dynamic content and deep linking for metrics and routing students to appropriate pages. We would also like to add additional error-checking and create more descriptive error messages for the user. Last but not least, we want mobile users to also experience the application; making Teacher Supply Donation mobile-responsive is the next top priority goal for our team.