# BC CANCER PROJECT PROPOSAL

**CS5500 - Group 4**

Dalin Wang
Meilin Niu
Minghe Hu
Rubing Li
Yuning Mu

Oct 21, 2024

# Project Overview

## Objective

- Enhance BC Cancer's donor data management by implementing a **modern, smart donor management system.**
- Enhance **coordination** among event managers, fundraisers, and coordinators.
- Improve **operational efficiency and effectiveness** in donor management and event planning.

## Deliverables

- Wireframe prototype and project proposal
- A comprehensive platform for managing the events and tasks

# TABLE OF CONTENTS

# 01.

# Problem Statement

Context, problem statement and target audience

# 01 Problem Statement

## Context

- **Donor contributions are essential** in supporting BC Cancer's efforts in cancer research and patient care.
- Current management methods are **inefficient and prone to errors.**
- Inefficient management **hampers fundraising efforts and staff coordination.**

## Problem

- **Lack of a centralized, smart** donor management system.
- **Challenges in coordinating** events and **tracking** donor interactions.
- Inefficiencies leading to **potential loss of donations** and engagement opportunities.

# TARGET AUDIENCE

- **Event managers** at BC Cancer.
- **Fundraisers and Coordinators**.
- **Donors** who will benefit from improved engagement.
- Ultimately, **cancer patients and research initiatives** supported by **increased funding.**

# 02.

# Potential Solutions

Exploring Three Solutions and Selecting the Best Fit

# POTENTIAL SOLUTION - A

## Enhance Existing Spreadsheets

**Tools Involved**:

- Excel, PowerBI

**Pros**:

- Quickly modify pre-existing files.
- Easy and comfortable for users to get started without a steep learning curve.

**Cons**:

- Limited scalability.
- Unstructured and inconsistent file management.

# POTENTIAL SOLUTION - B

**Leverage Real-time Editing with donor system**

- Allows multiple users to edit the data at the same time
- Implements CRUD on donor system

**Tools Involved**:

- ELK

**Pros**:

- Edit the table more efficiently

**Cons**:

- Limited budget
- Requires additional developers

# POTENTIAL SOLUTION – C

**Develop a Tailored Spreadsheet Management System**

- Offer user-friendly interfaces and enhanced control over event and task management

**Tools Involved**:

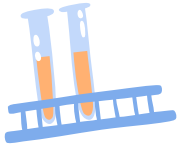
- React for front-end development; Node.js for back-end development; MySQL for structured data storage

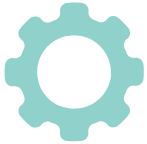
**Pros**:

- Full control over file structure and data management.
- Can be tailored to specific user needs, offering advanced tools for data analysis and reporting.

**Cons**:

- Longer implementation time.
- Requires training for users to adapt to the new system.





# 03. Wireframe Prototype

Process Flow and Interface Sketches

# Key Features

## Add New Event

Generate a attendee list in a convenient and effective way

## Event Editing Workbench

Avoid the troubles of using traditional Excel worksheets

## View Event Details

Record the process of decision making

# PROCESS FLOW OF SYSTEM

01

## Get a Preliminary List

Select the criteria to generate a preliminary list

02

## Review and Edit

Open for review and edits by the personnel

03

## Approve the Final List

Submit the final decision

04

## Extract Sub-Lists

Split based on notification preferences e.g. Newsletters, Emails

# PROCESS FLOW OF SYSTEM

01

## Get a Preliminary List

Select the criteria to generate a preliminary list

02

## Review and Edit

Open for review and edits by the personnel

03

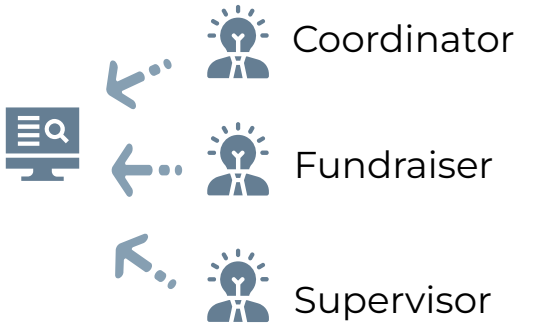
## Approve the Final List

Submit the final decision

04

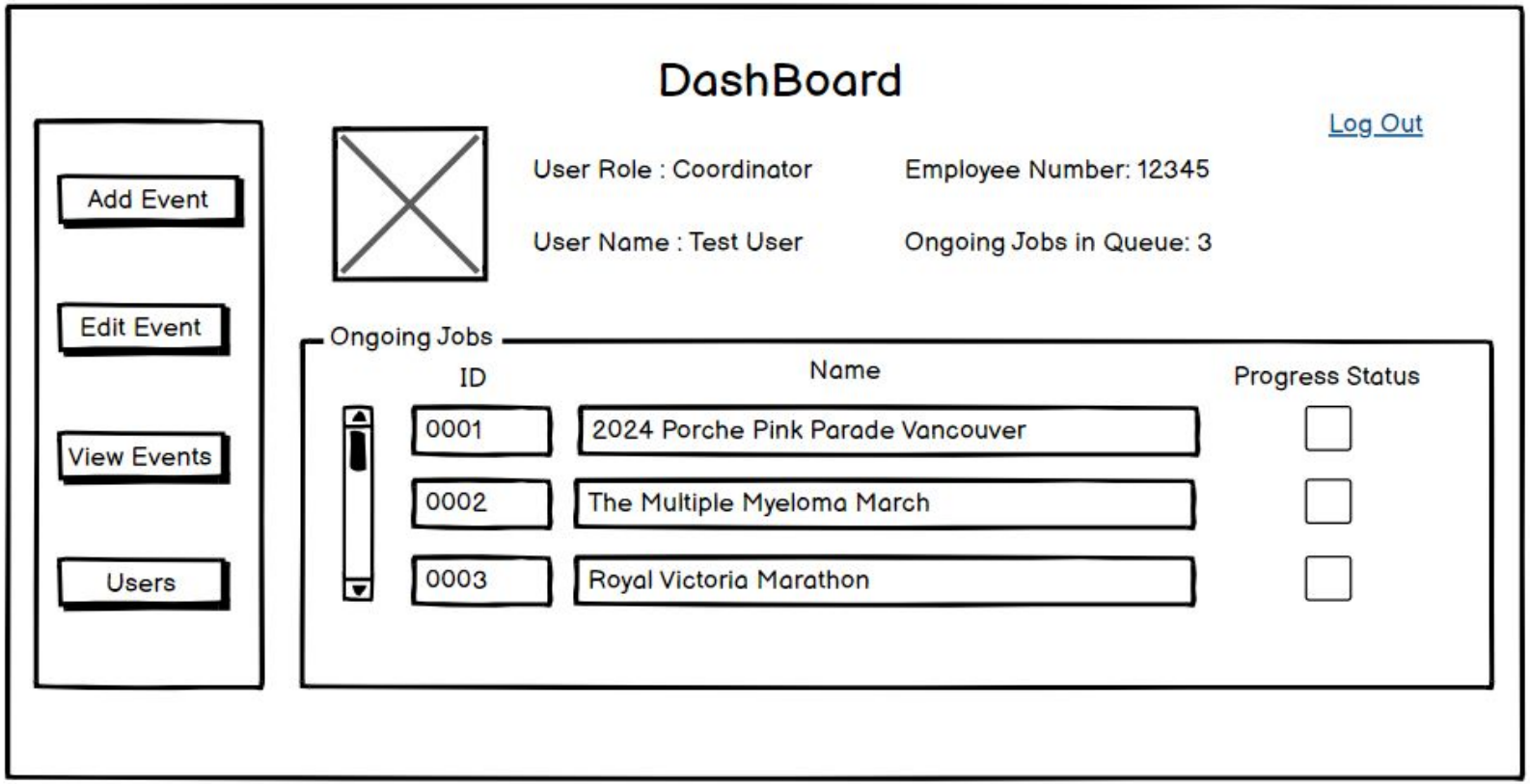
## Extract Sub-Lists

Split based on notification preferences e.g. Newsletters, Emails

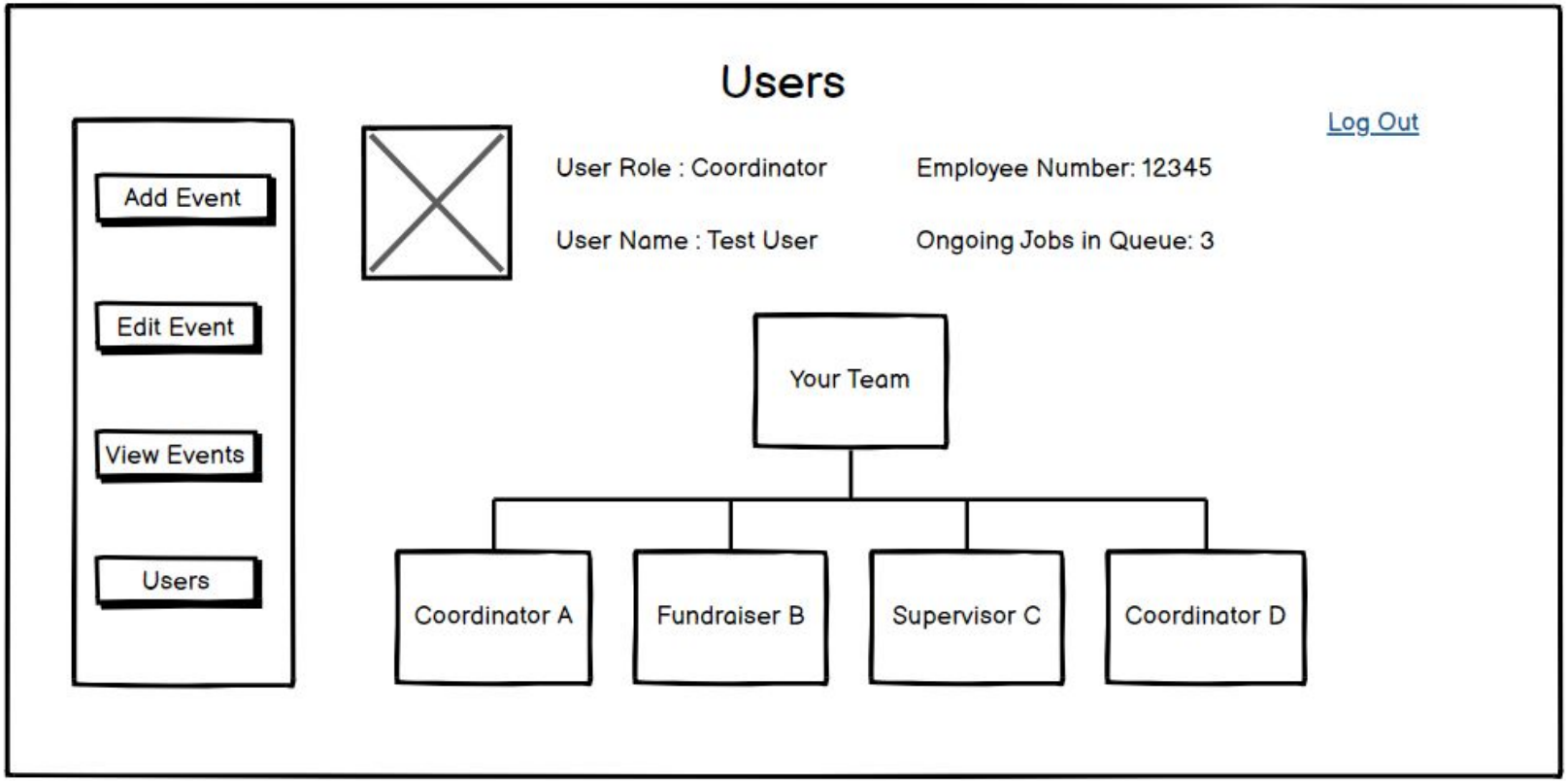



All the edit history and reasons for editing will be recorded.

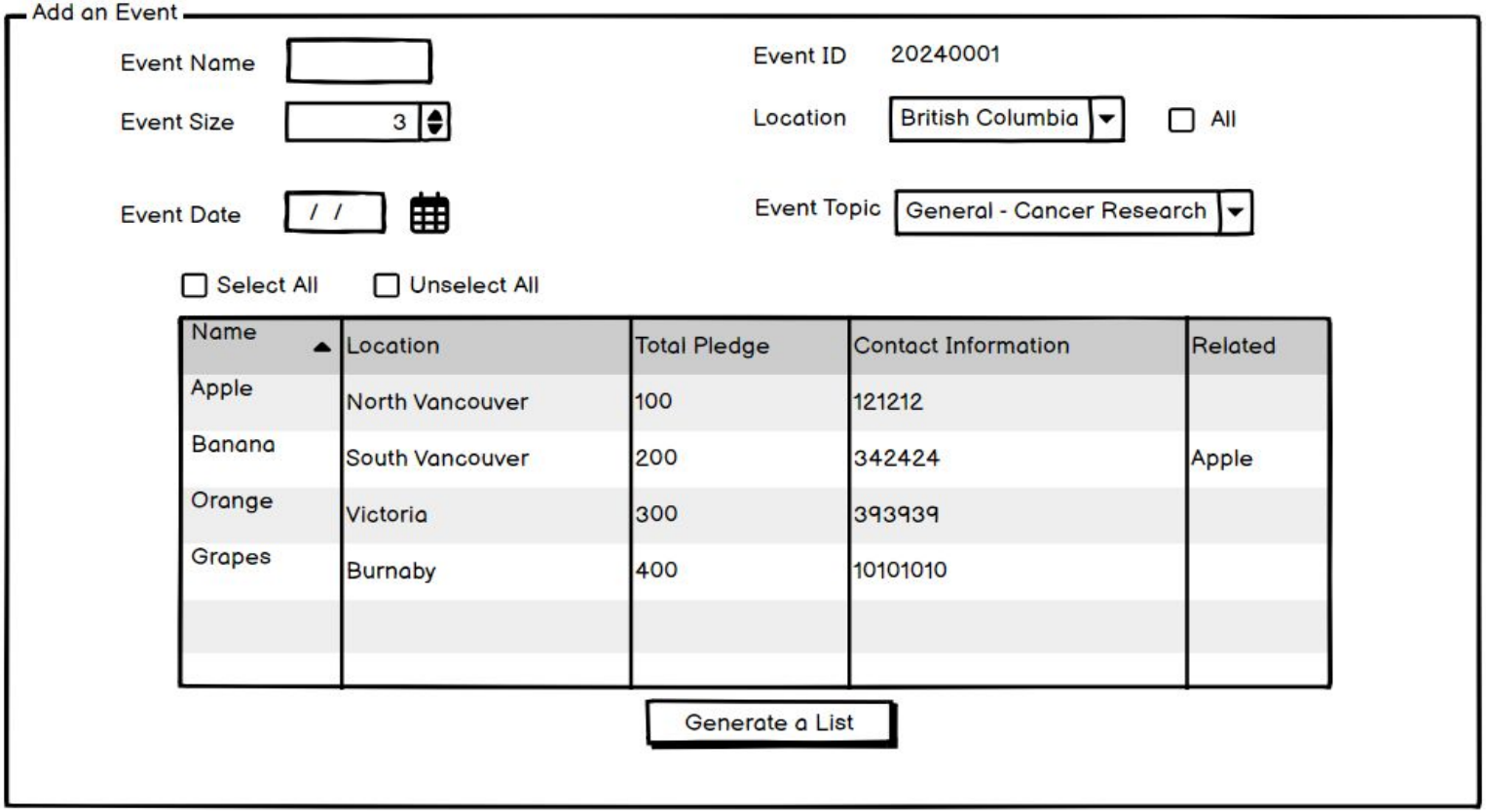
# User Interface - Dashboard



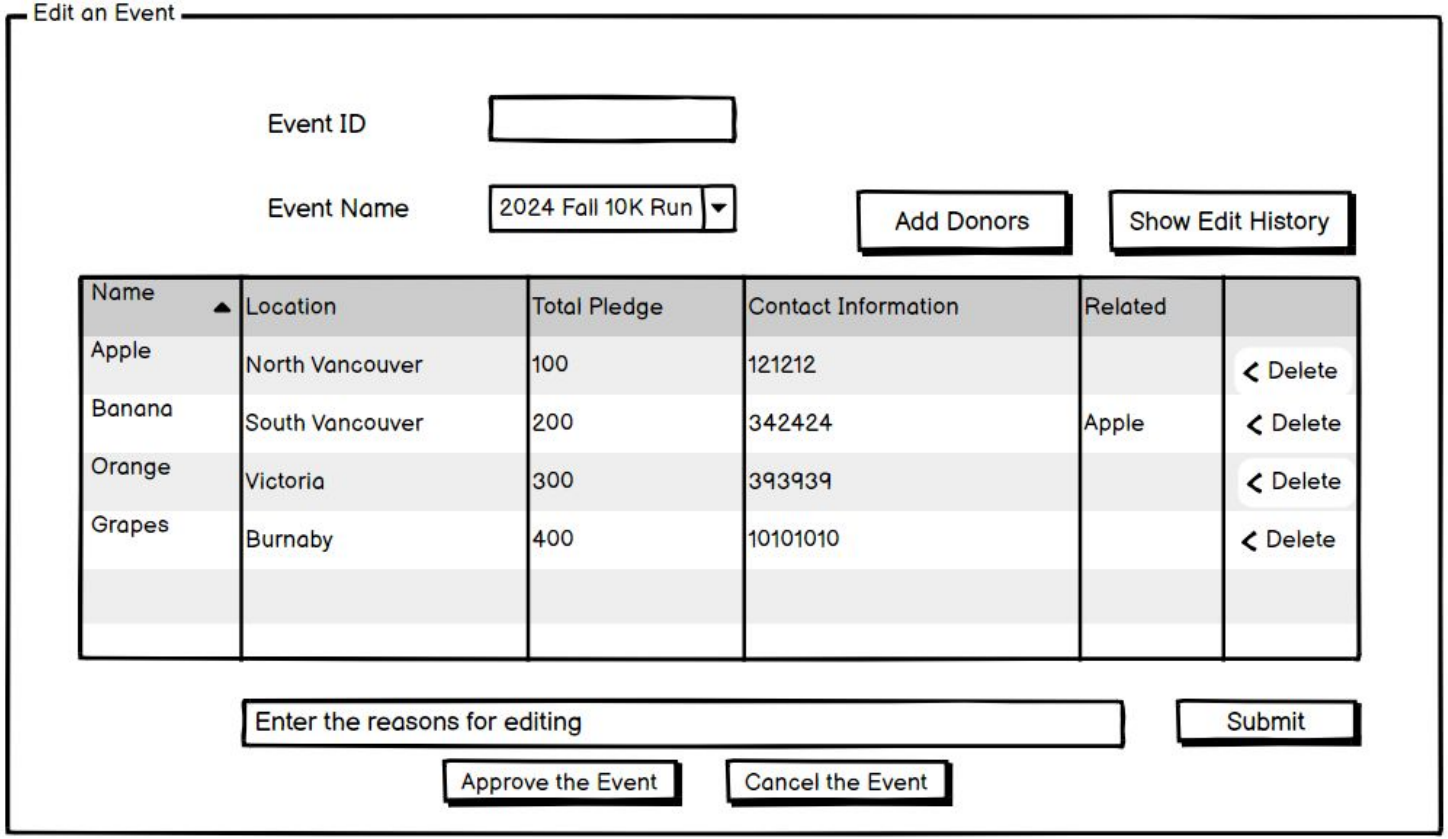
# User Interface - Users



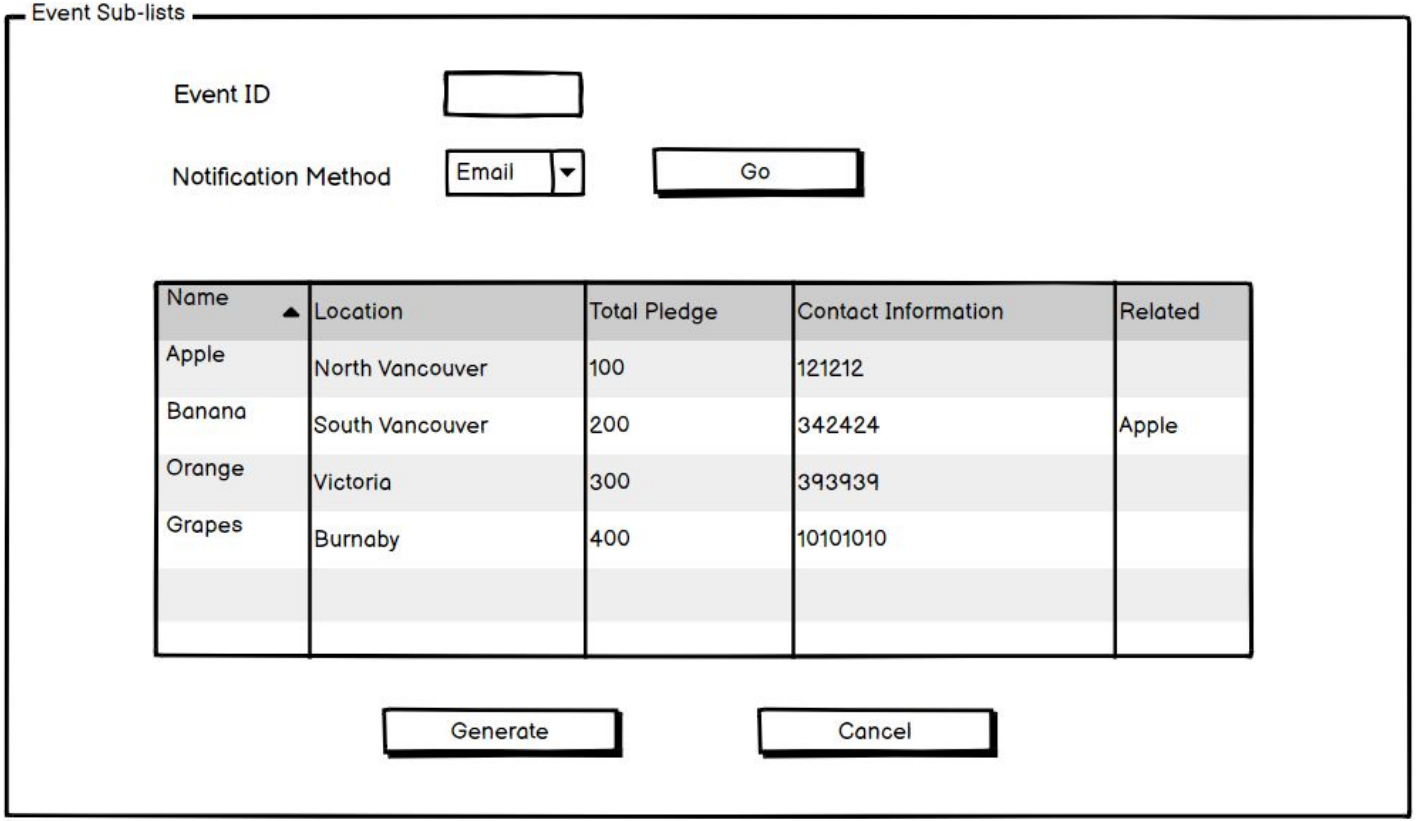
# User Interface - Add Event



# User Interface - Edit Event



# User Interface - Extract Sub-Lists



# 04.

# TECH STACK & CHALLENGES

Technologies involved, strategies and uncertainties

# TECH STACK

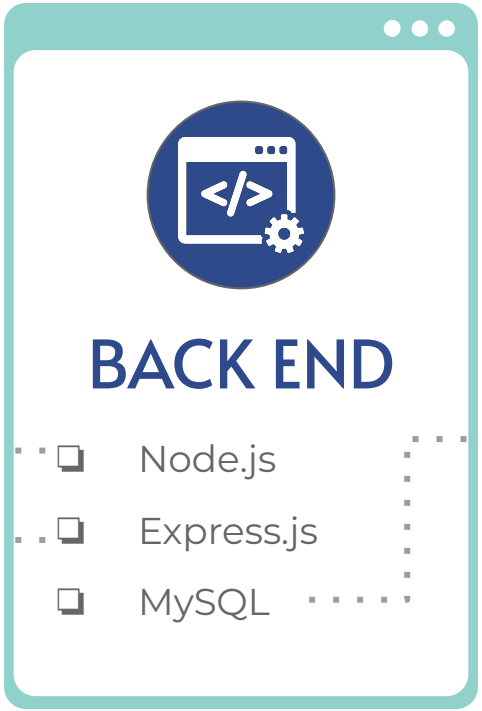
## FRONT END

- ❑ React.js

- ❖ Our team is most comfortable with React
- ❖ It enables dynamic, responsive interfaces with reusable components.

# TECH STACK

- Integrate smoothly with the rest of our stack
- Efficiently manage routing and server-side logic, keeping everything streamlined




- Relational database that is space-saving, organized, and scalable for our needs.

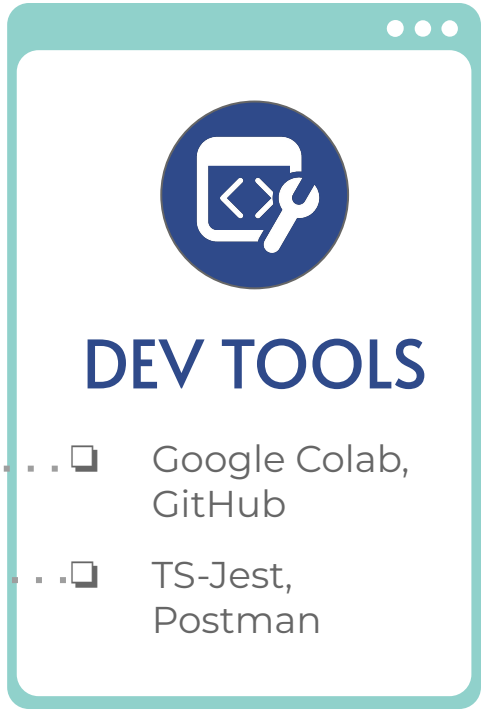
# TECH STACK

## Collaboration

- **GitHub** for version control and collaboration on the majority of our codebase
- **Google Colab** as a shared environment for research and testing scripts

## Testing

- **Jest** for comprehensive testing
- **Postman** ensures smooth API communication between frontend and backend



# TECH STACK

## FRONT END

- ❏ React.js

## BACK END

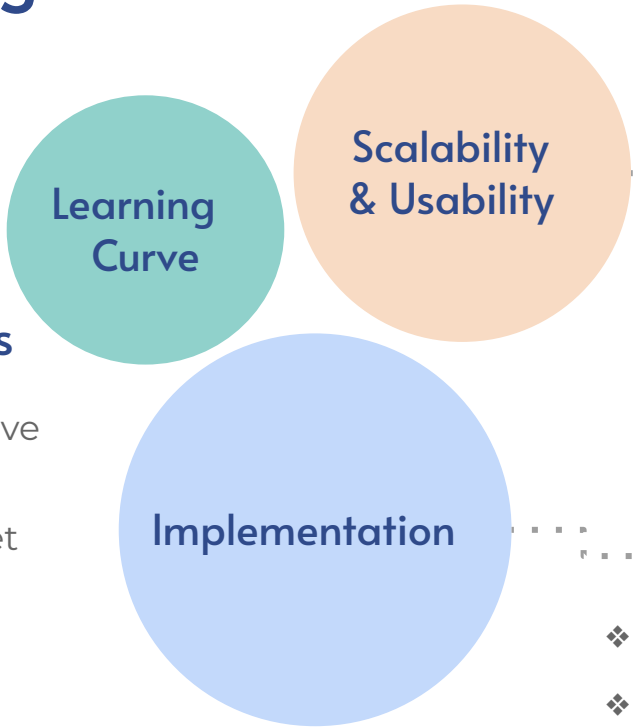
- ❏ Node.js
- ❏ Express
- ❏ MySQL

## DEV TOOLS

- ❏ Google Colab, GitHub
- ❏ ts-Jest, Postman

TS **TypeScript** as our main language: better code **quality**, static **typing**, **maintainability**

# CHALLENGES




## Tackle Unfamiliar Tools

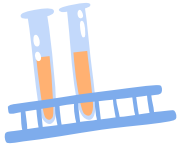
- Focused and collaborative learning.
- Hands-on practice to get familiar with TypeScript and testing tools.

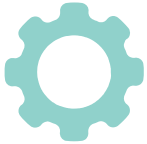
## Ensure User Experience

- Modular architecture for future expansions
- Optimizing server-side logic for faster response times

## Complex Features

- Role-based access control
- Seeking references from existing frameworks





# 05. FUTURE WORK

Our plans and future steps


# Development plan

## FRONT END

- UI

## Final Application

## BACK END

- Database

Currently our development plan is to separate two different groups: one for backend and one for front end.

# 06.

# CONCLUSION

# CS 5500 - Group 4

Dalin Wang

Meilin Niu

Minghe Hu

Rubing Li

Yuning Mu

# THANK YOU !