Project Overview

This project is a centralized form-filling application that allows users to access different forms from different organizations with one website. This project involves allowing different organizations to create forms and allow users to fill out created forms. In addition to this, it gives users a way to see all the forms they have filled in before for tracking purposes. The major goal of this project is to simplify the current system of filling out forms and make it much easier to create different forms and also to easily fill out forms.

Bio

We are the Former Wyoming Business Registration Team (name subject to change), formed to make a functional system to allow centralized coordination for diverse kinds of groups. We are comprised of three undergraduate University of Wyoming students with a passion for computing.

Fortune Nwokejiobi – Front-end design

Miles Hays – Backend logic

Lucas Eifert – Database management

MVP Functional Requirements / Use Cases

First, create an account or log in.

A user then has a panel, from which they can go to their profile, list of member organizations, or create an organization. They also have a box to search for public organizations.

In their personal profile, a user can edit information and accept invites from organizations.

In the personal information dashboard, a user is shown a short form which comprises of text fields containing basic information which we expect to be common information across organizations. Examples of this include name, address etc.

In the create page, a user gives initial organization information, and is directed to the organization's admin page.

In the organization page, the user has a panel, from which they can choose to go to their organizational profile or view available forms. If they are an admin, they have the option to view an administrative page.

In the admin page, the user has access to all forms, where they can view or download responses. They can also invite users to join, grant admin privileges, or create a form.

Form builder to be determined over break.

Systems

Form Filling

- Form spec file
    - JSON

    - Contains enough data to build HTML components
    - Contains enough data to link to common data engine
- Input validation
    - Validate input as users input (basic HTML form type) and backend input validation before inserting
    - Field type specified in form spec file
- Draft filings
    - Not visible to organization
    - Can be saved and edited until actual submission

Common Data

- Personal information dashboard, includes name, address, etc.
- Organizations include their own personal information dashboard
- Data entered within form will be assigned a name
- Data with the same name will auto-fill in later forms once entered

Authentication

- Interact with 3rd-party login APIs

File System

- Store JSON for blank form templates
- Store JSON for completed forms
- Store JSON for in-process forms
- Store files (such as images) uploaded from forms

Database

- MySql
    - Organization – ID, Name, Users, Admins, Links to form templates, Forms
    - User – ID, Name, DOB, SN, Address, Phone number, Link to in-process forms, member organizations, admin organizations, Organization specific information
    - Form – ID, Organization, Name, User, Category, Date completed, attachments(links to images, etc.)

Website

- Express + React

Form builder

- Create fields
    - Data type
    - Data name (for common data engine)
- Form Management
    - View and download response data

Milestones

Complete Design Specification

- UI drawings for all pages
- Database diagram for complete system

User Authentication

- Login page
- Credential management
- Security

Form System

- JSON specification for form
- Translate JSON to React components
- Render Form
- Store entire submission on submit

Common Data Engine

- Personal information dashboards
- Extract named data from form when updated, update individual database entries
    - Confirm update first?
- Pull information from database on form load

Form Builder

- Read & write JSON according to form spec
- Graphical interface to assemble forms
- Define validation rules for any field

Tasks

Initial configuration – Fortune

Front-end conceptual design – Fortune

Front-end page implementations – Fortune

Database specification – Lucas

Database creation – Lucas

Database API – Lucas

File system integration – Lucas

Form JSON specification – Miles

Login system – Miles

Form rendering – Fortune

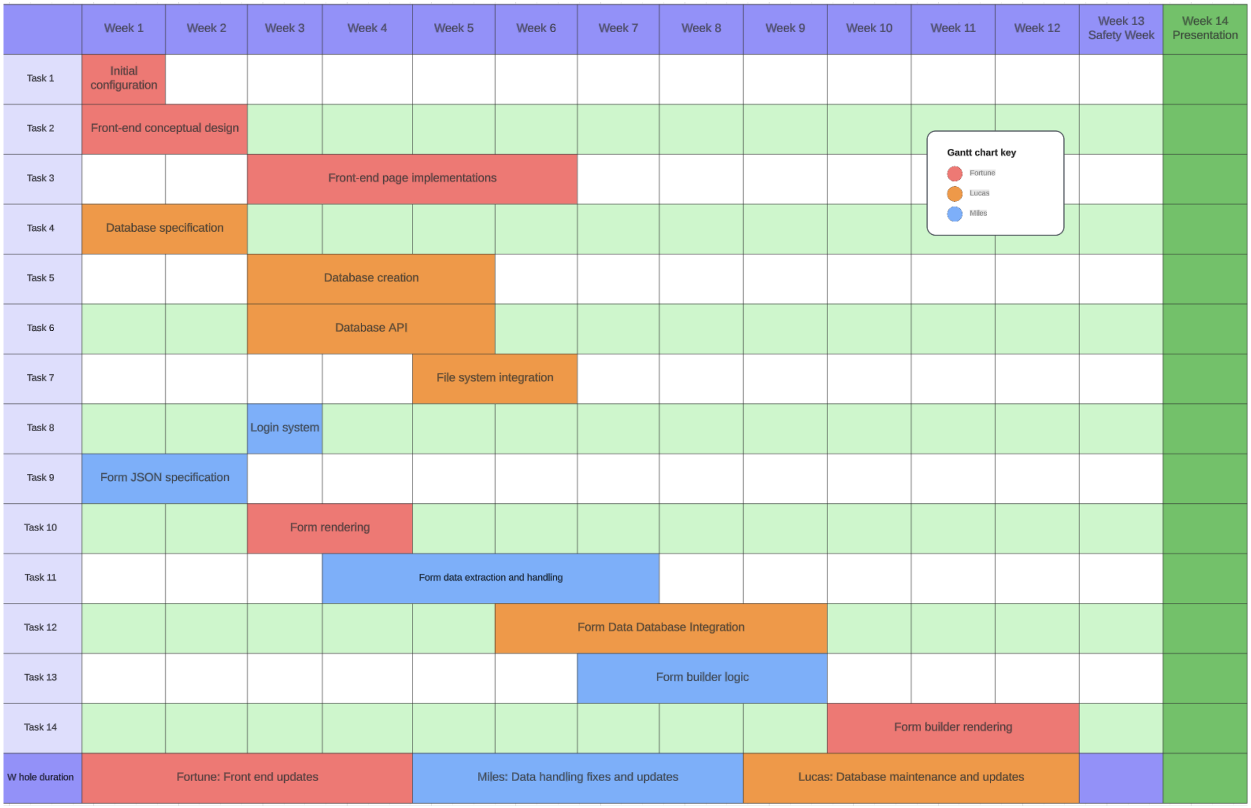
Form data extraction and handling – Miles

Form data database integration – Lucas

Form builder logic – Miles

Form builder rendering – Fortune

Timeline



	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13 Safety Week	Week 14 Presentation
Task 1	Initial configuration													
Task 2	Front-end conceptual design													
Task 3			Front-end page implementations											
Task 4	Database specification													
Task 5			Database creation											
Task 6			Database API											
Task 7					File system integration									
Task 8			Login system											
Task 9	Form JSON specification													
Task 10			Form rendering											
Task 11				Form data extraction and handling										
Task 12						Form Data Database Integration								
Task 13							Form builder logic							
Task 14										Form builder rendering				
W hole duration	Fortune: Front end updates				Miles: Data handling fixes and updates				Lucas: Database maintenance and updates					

Gantt chart key

● Fortune

● Lucas

● Miles

(Somewhat optimistic)

Stretch Goals

- Grant access for different users to different forms within a single organization
- Form urgency
- Varying public-ness of organizations (i.e. choose between public list or invite only)
- Better support for corporate end users
- Fancier form builder
- 3rd-party authentication integration