enjerneering UI Builder

Pierson, Jake, Jadi, Payton

Jadi

Jadi ->

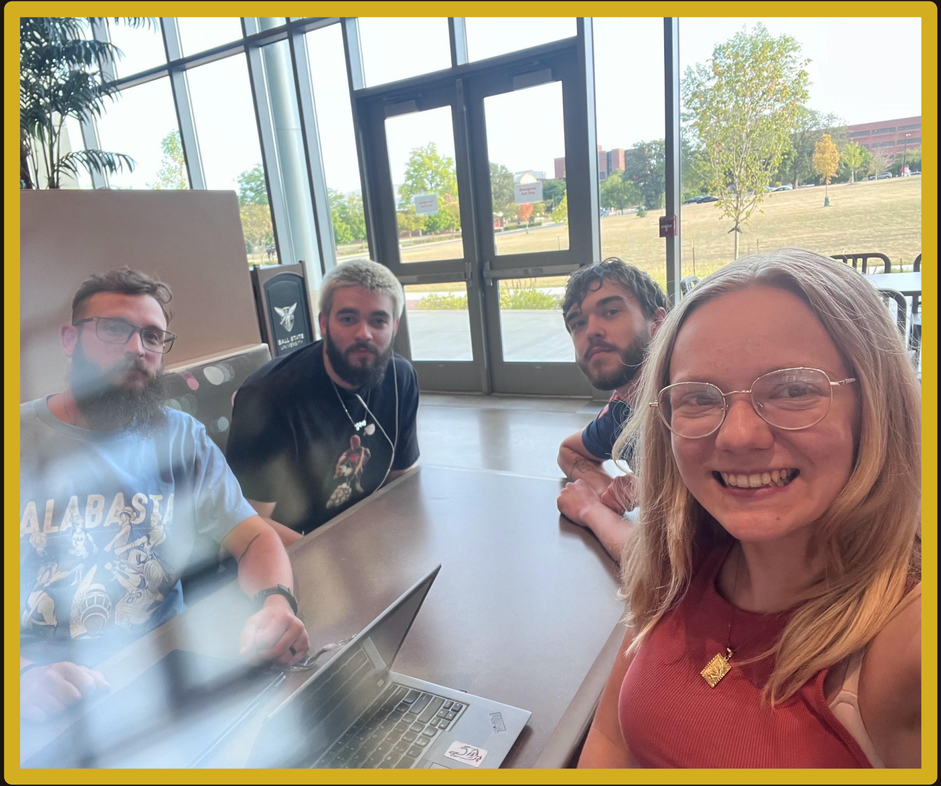
Team Member Information

Jadi - Specializes in making games and interactive software

Jake - Prefers frontend and UI development

Pierson - Enjoys backend coding portions of projects, Seahawks Fan (unfortunately)

Payton - Enjoys backend programming



Jadi

Pierson ->

Something Interesting...

Pierson got a kitten!

Pierson

Jadi ->

Client Information

Jerrod Smith, Owner of EnJerneering

- Flexware Innovation Software Engineer
- This is a side project outside of his typical scope of work
- Outsourced a design team overseas who have set up the repository, built the UI, and implemented some of the bare bones of the project

Business Requirements

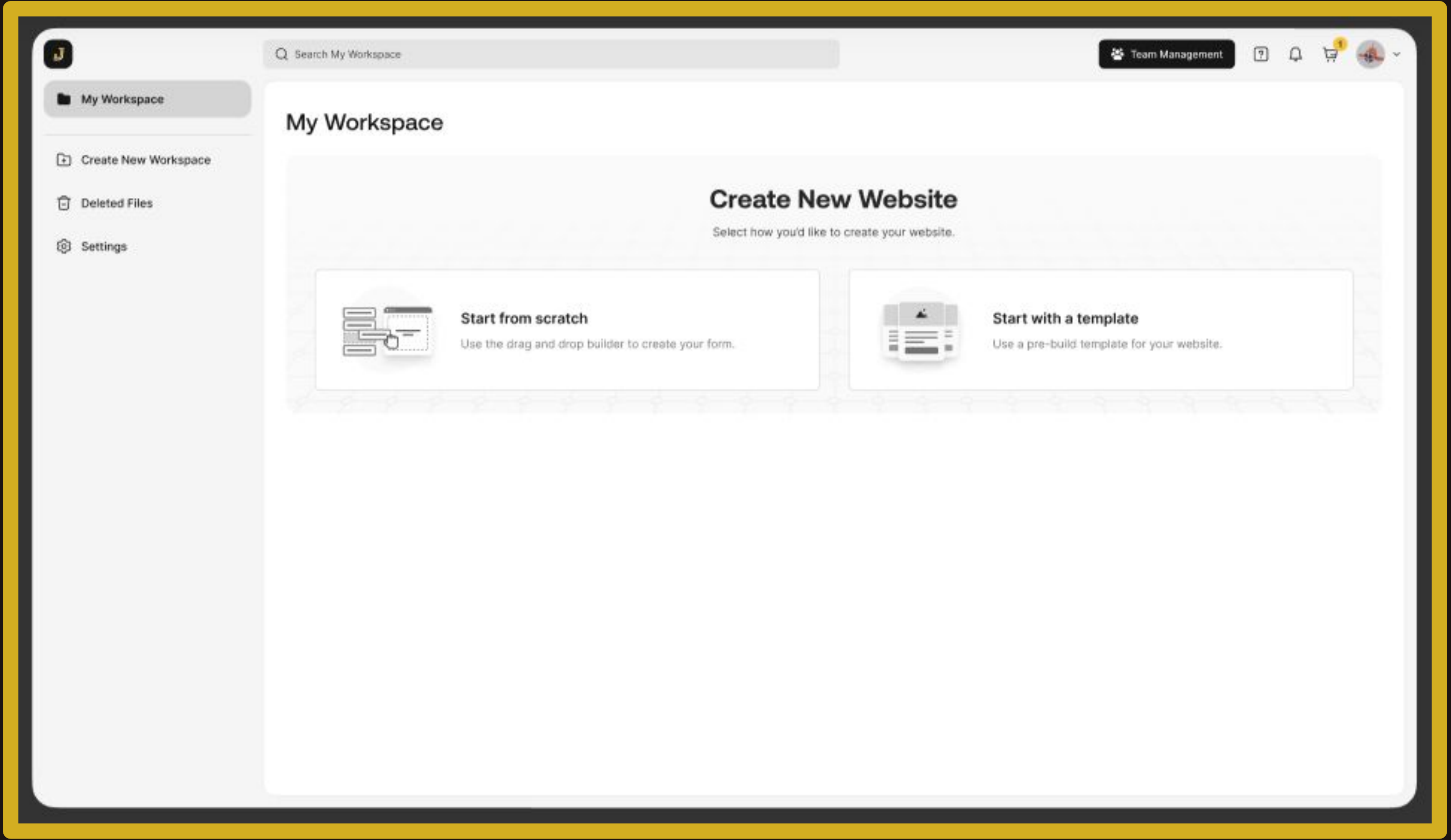
BR1:

The platform will allow freelance web designers to create and manage multiple websites through a drag-and-drop interface.

BR2:

The platform will allow freelance web designers to export and deploy their projects in Next.js.

Prototype



My Workspace

Create New Website

Select how you'd like to create your website.

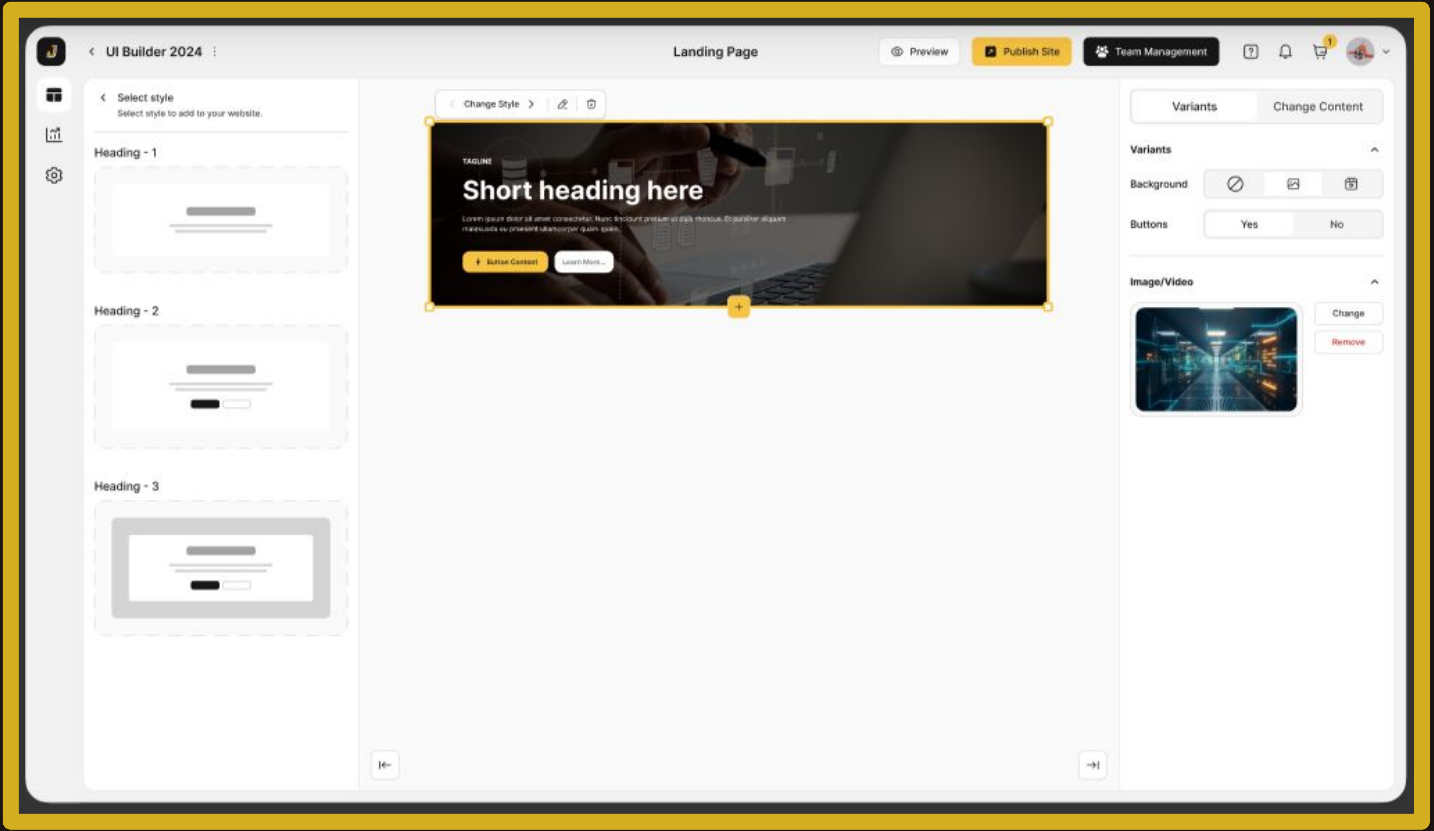
Start from scratch

Use the drag and drop builder to create your form.

Start with a template

Use a pre-build template for your website.

Prototype



Jadi

Payton ->

Use Cases

- UC1: Streamlined Web Development Process
- UC2: Customization and Full Code Ownership for Clients
- UC3: SEO Optimization During Development
- UC4: Website Creation for Marketing Clients
- UC5: Enhanced Style Customization for Freelancers
- UC6: Smart Page Management for Continuous Updates
- UC7: Specialized Website Development for Health and Wellness Clients
- UC8: Efficient Content Management for Niche Clients

Payton

Pierson ->

Requirements

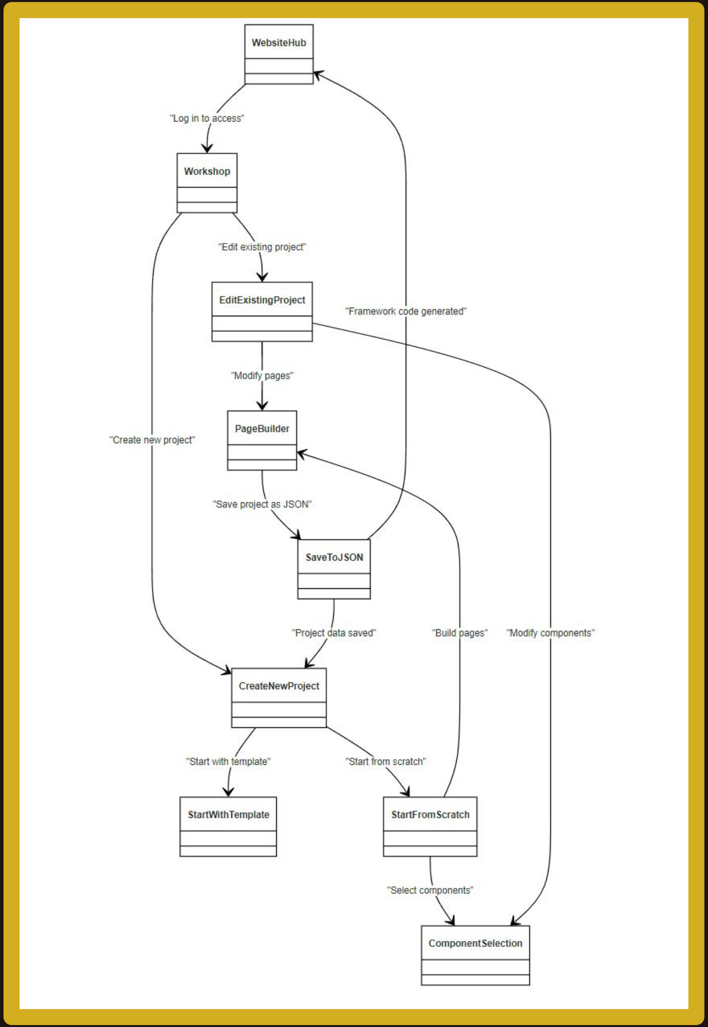
- FR1: A User will be able to create a project to design their website, which can be saved for future updating (UC4) [high]
- FR2: The system will allow a user to edit and update previously saved projects (UC6) [high]
- FR3: The setup page for a new project will allow the user to input key information for their web project, which can be used to create a base template for their project (UC5) [high]
- FR4: Component options can be dragged and dropped by a user into the UI builder to design the layout for their project (UC1) [high]
- FR5: Projects can include multiple pages that can follow the same design layout and have individual functionalities (UC8) [medium]
- FR6: Projects can be exported to a generalized JSON format, which can be converted into next.js to create the final website (UC2) [medium]
- FR7: Pre-designed templates are available for users when they create their project (UC5) [low]

- NR1: User's projects and information are safely and reliably stored to ensure their work is not lost (UC7) [high]
- NR2: UI builder is optimized and organized as to not hinder users' workflows (UC3) [low]

Pierson

Payton ->

Domain Model



WebsiteHub

"Log in to access"

Workshop

"Edit existing project"

EditExistingProject

"Framework code generated"

"Modify pages"

PageBuilder

"Create new project"

"Save project as JSON"

SaveToJSON

"Project data saved"

"Build pages"

"Modify components"

CreateNewProject

"Start with template"

"Start from scratch"

StartWithTemplate

StartFromScratch

"Select components"

ComponentSelection

Payton

Jadi ->

Tech Stack

Next.js/ Typescript

"It doesn't let you write bad JavaScript."

Prisma ORM

Code-first approach to databases

Stays in the repository as the frontend code

Supabase

Database service that makes database building easier

Clerk Authenticator

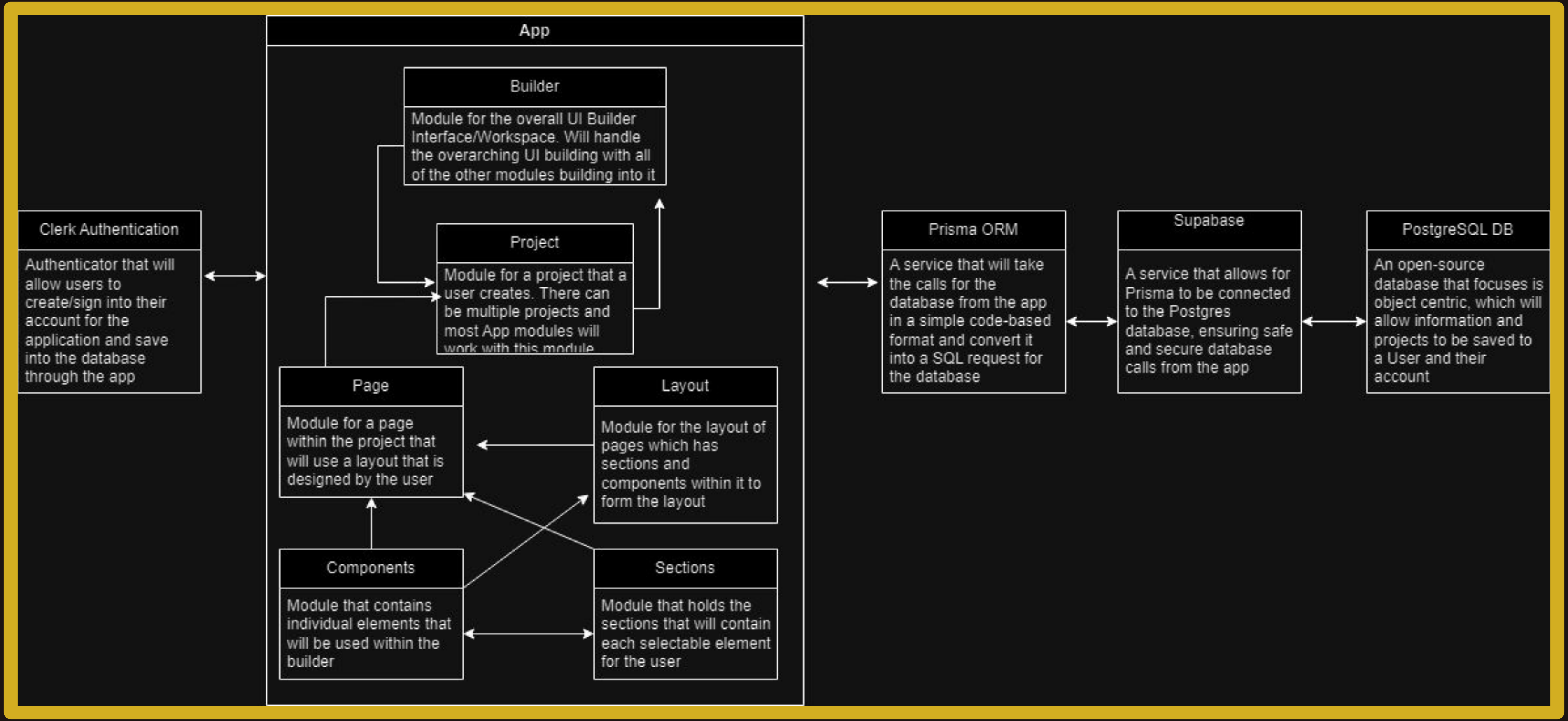
Avoid security certificates

Avoid building custom login service

Jadi

Pierson ->

Architecture



App

Builder

Module for the overall UI Builder Interface/Workspace. Will handle the overarching UI building with all of the other modules building into it

Project

Module for a project that a user creates. There can be multiple projects and most App modules will work with this module

Page

Module for a page within the project that will use a layout that is designed by the user

Layout

Module for the layout of pages which has sections and components within it to form the layout

Components

Module that contains individual elements that will be used within the builder

Sections

Module that holds the sections that will contain each selectable element for the user

Clerk Authentication

Authenticator that will allow users to create/sign into their account for the application and save into the database through the app

Prisma ORM

A service that will take the calls for the database from the app in a simple code-based format and convert it into a SQL request for the database

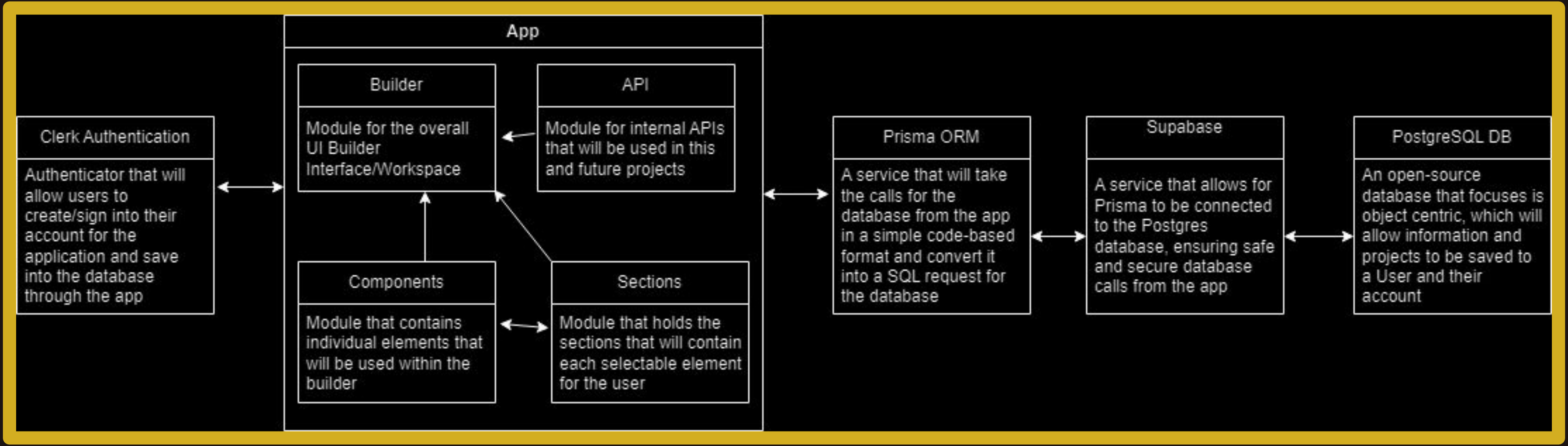
Supabase

A service that allows for Prisma to be connected to the Postgres database, ensuring safe and secure database calls from the app

PostgreSQL DB

An open-source database that focuses is object centric, which will allow information and projects to be saved to a User and their account

Architecture Before Feedback



Clerk Authentication

Authenticator that will allow users to create/sign into their account for the application and save into the database through the app

App

Builder

Module for the overall UI Builder Interface/Workspace

API

Module for internal APIs that will be used in this and future projects

Components

Module that contains individual elements that will be used within the builder

Sections

Module that holds the sections that will contain each selectable element for the user

Prisma ORM

A service that will take the calls for the database from the app in a simple code-based format and convert it into a SQL request for the database

Supabase

A service that allows for Prisma to be connected to the Postgres database, ensuring safe and secure database calls from the app

PostgreSQL DB

An open-source database that focuses is object centric, which will allow information and projects to be saved to a User and their account

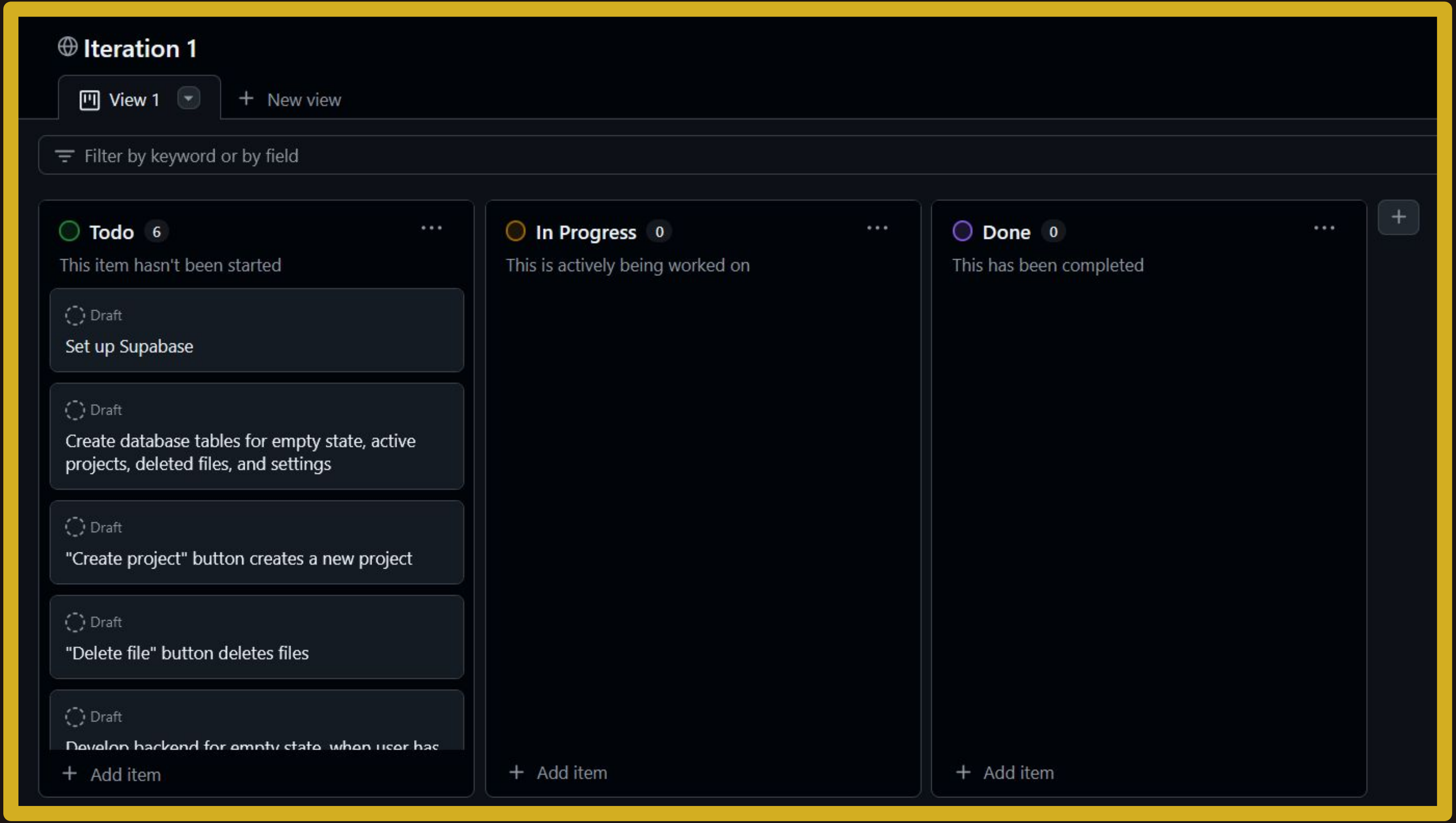
Pierson

Jadi ->

First Iteration Features

- Set Up Backend Infrastructure Using Supabase
  - Implement the backend database using supabase with pre-defined
  - Connect frontend forms with data storage and retrieval in supabase
- Connect Most Important UI Elements to Backend
  - Link buttons (like ‘Create Project’ and ‘Delete Project’) to Supabase using API for data retrieval and submission
- Project Setup
  - Implement “Existing Website” form functionality
    - Add logic to ask if user has existing website
    - Capture website details if user has existing website
    - Validate and save the data in Supabase for use in project creation
  - Ensure correct formatting for fields like Domain or URL

First Iteration Features



Iteration 1

Todo 6
This item hasn't been started

- Draft
  Set up Supabase
- Draft
  Create database tables for empty state, active projects, deleted files, and settings
- Draft
  "Create project" button creates a new project
- Draft
  "Delete file" button deletes files
- Draft
  Develop backend for empty state, when user has

In Progress 0
This is actively being worked on

Done 0
This has been completed

Q & A

Questions about…

EnJerneering UI Builder

Architecture

Tech Stack

Domain Model

Requirements

First Iteration

Anything else?