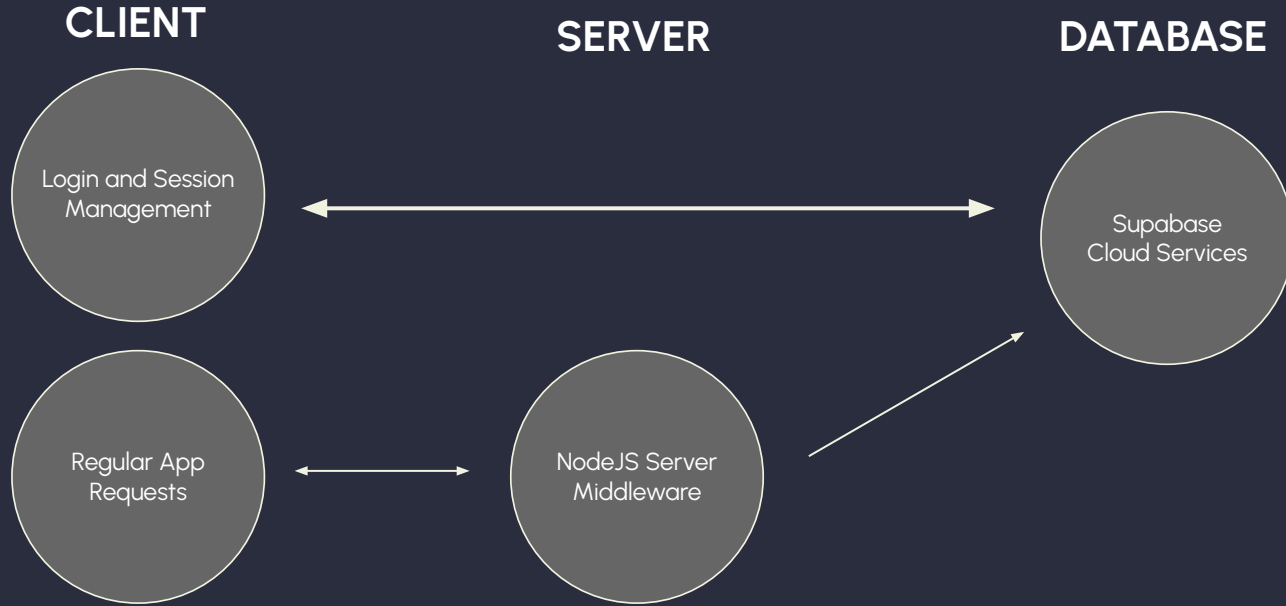# Midterm Project Update

Matt Smith
Jaden Henderson
Josh Burgenmeyer

Project Overview

- **Specialty Coffee Brewing Guide**

- **Core Features:**
  - Coffee inventory
  - Recipe creation
  - Brew timer that displays your selected recipe
  - Dashboard to view coffee consumption insights

- **Technologies**
  - React, Node, Express, Postgres
  - Progressive web app (PWA)

Task Number	TASK TITLE	START DATE	DUE DATE	Completion Status
1	Project Conception and Initiation	9/8/2025	9/27/2025	
1.1	Group Formation			100%
1.2	Project concept ideation			100%
1.3	Class Diagram			100%
1.4	Figma UI Design			100%
1.3	User Flow Diagram			100%
1.4	Sequence Diagram			100%
1.5	Use Case Diagram			100%
1.6	JIra Backlog Fill			50%
2	Implementation	9/28/2025		
2.1	Frontend Scaffolding			100%
2.2	Backend Scaffolding			100%
2.3	Supabase Table Creation			100%
2.4	Recipe Creation UI			50%
2.5	API Endpoints for Recipe Creation			0%
2.5	Login/Sign Up UI Forms			100%
2.6	Supabase User Authentication			100%
2.7	Coffee Bag Entry UI Creation			75%
2.8	Timer UI Creation			0%
2.9	API Endpoints for Coffee Bag Creation			0%
2.1	Brew Result UI Creation			75%
2.11	Data Visualization UI			0%
2.12	Remaining API Endpoints			0%
3	Testing & Revision			
3.1	Frontend Testing			0%
3.2	API Endpoint Testing			0%
3.2.1	Cross Browser Testing			0%
3.2.2	Performance Testing			0%

Gantt chart columns: PHASE ONE (WEEK 1, WEEK 2, WEEK 3), PHASE TWO (WEEK 4, WEEK 5, WEEK 6, WEEK...), each with days M T W R F.

Network Structure

CLIENT SERVER DATABASE

Login and Session Management

Supabase Cloud Services

Regular App Requests

NodeJS Server Middleware

Sends POST requests from the frontend to an express/node JS server that edits a Supabase db

Design Decisions

Straightforward network structure

Data is sent to a single handler that parses it and edits the db accordingly.

Security handled by established services

Login/Sign Up authorizations and session management tokens are handled completely by Supabase.

Database handled by cloud provider

Supabase provides a simple and flexible way to store and modify data without having to worry too much about syntax.

Example Network Request

CLIENT

Request Headers ☐ Raw

Accept:	*/*
Accept-Encoding:	gzip, deflate, br, zstd
Accept-Language:	en-US,en;q=0.9
Authorization:	Bearer

eyJhbGciOiJIUzI1NiIsImtpZCl6ImVwSnJXamxKS0x2N0E3QlIiLCJ0eXAiOiJKV1QifQ.eyJpc3MiOiJodHRwczovL2Z5dXFmeW5tYXl6anh3Rm5sb05hU3AwRlloYW5h
XNILmNvL2F1dGgvdjEiLCJzdWIiOil2ZTQ4NmZkMi05NjZhLTRhNjYtOWE2Zi0zODQ3NDA5MWQ4NTliLCJhdWQiOiJhdXhRoZW50aWNhdGVkliwiZXhwljoxN
zYwNDg4ODMzLCJpYXQiOjE3NjQ0ODU1MzMsImVtYWlsljoiamFkZW5AZXhhbXBsZS5jb20iLCJwaG9uZSl6liIsImFwcF9tZXRhZGF0YSl6eyJwcm92aWRlciliOil
mVtYWlsliwicHJvdmlkZXJzIjpblmVtYWlsll19LCJ1c2VyX21ldGFkYXRhljp7ImVt
YWlsljoiamFkZW5AZXhhbXBsZS5jb20iLCJlbWFpbF92ZXJpZmllZCl6dHJ1ZSwicGhvbmVfdmVyaWZpZWQiOmZhbHNILCJzdWliOil2ZTQ4NmZkMi05NjZhLT
RhNjYtOWE2Zi0zODQ3NDA5MWQ4NTlifSwicm9sZSl6ImF1dGhlbnRpY2F0ZWQiLCJhYWwiOiJhYWwxIiwiYW1yljpbeyJtZXRob2QiOiJwYXNzd29yZCIsInRpbWVzdGFtcCl6MTc2NDQ4NTUzM319LCJzZXNzaW9uX2lkljoiZWQyMGJiNjAtMzY3MS00NTIzM31dLCJzZXNzaW9uX2lkljoiZWQyMGJiNjAtMzY3MS00NTIzM31dLCJzZXNzaW9uX2lkljoiZWQyMGJiNjAtMzY3MS00NTIzMzhmODlkMWZiliwiaXNfYW5vbnltb3VzljpmYWxzZX0.L82yoZdzxiUVYKp44WhmGmvsoPYyKM4RmSmyAG4ojVI

|	
Connection:	keep-alive
Content-Length:	160
Content-Type:	application/json

Request Payload   view source

```
{name: "Blonde Roast", roaster: "Peet's C
  name: "Blonde Roast"
  origin: "brazil"
  process: "Grinding"
  roastDate: "2025-10-14T04:00:00.000Z"
  roaster: "Peet's Coffee"
  variety: "pacas"
  weight: "150"
```

SERVER

```
app.post("/api/addCoffee", requireUser, async (req, res) => { // POST addCoffee
  var parsed = FormDataSchema.safeParse(req.body);
  console.log(parsed);

  if (!parsed.success) {
    return res.status(400).json({ error: 'Invalid data', details: parsed.error.flatten() });
  }

  try {
    const userId = req.user.id;

    const payload = { ...req.body, user_id: userId };

    const { data, error } = await supabaseAdmin.from('coffeeBag').insert(payload).select('*').single();

    if (error) {
      console.error('Supabase insert error:', error);
      return res.status(500).json({ error: 'Database error' });
    }

    return res.status(201).json({ ok: true, coffee: data });
  } catch (err) {
    console.error('Server error:', err);
    return res.status(500).json({ error: 'Server error' });
  }
});
```

DATABASE

id int8	created_at timestamptz	weight int8	variety text	roastDate date
1	2025-10-14 00:26:20.369634+0	150	maragogipe	2025-10-09
2	2025-10-14 00:28:23.743078+0	175	pacamara	2025-10-09
3	2025-10-14 23:42:12.84025+00	150	pacas	2025-10-14

6

Figma Diagrams

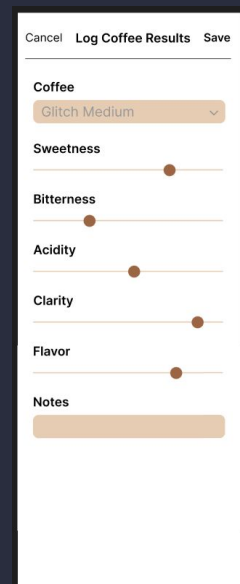## Recipe Components
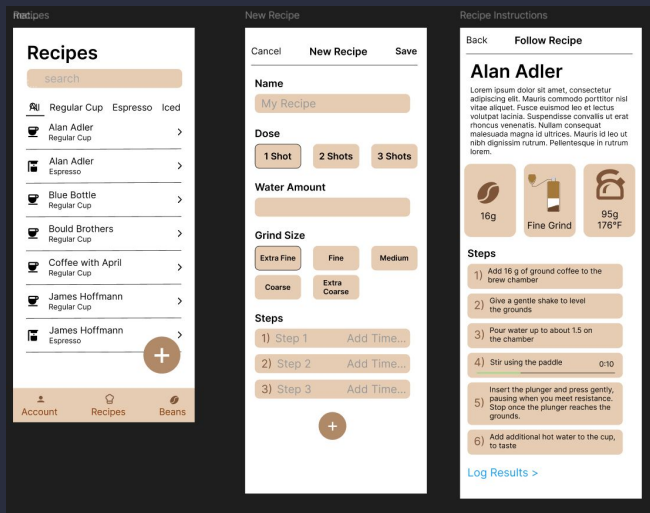
Login Components
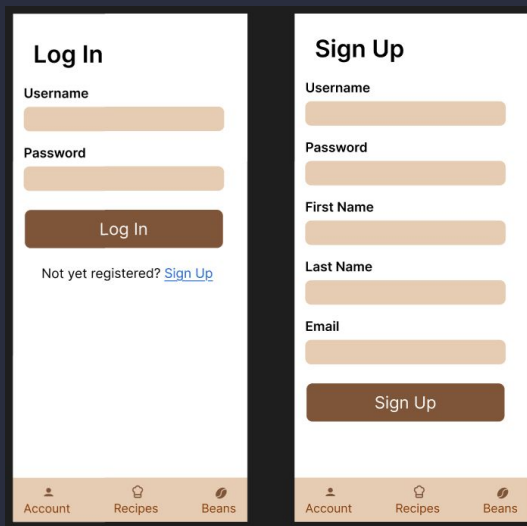
Logging

Recipe React Components
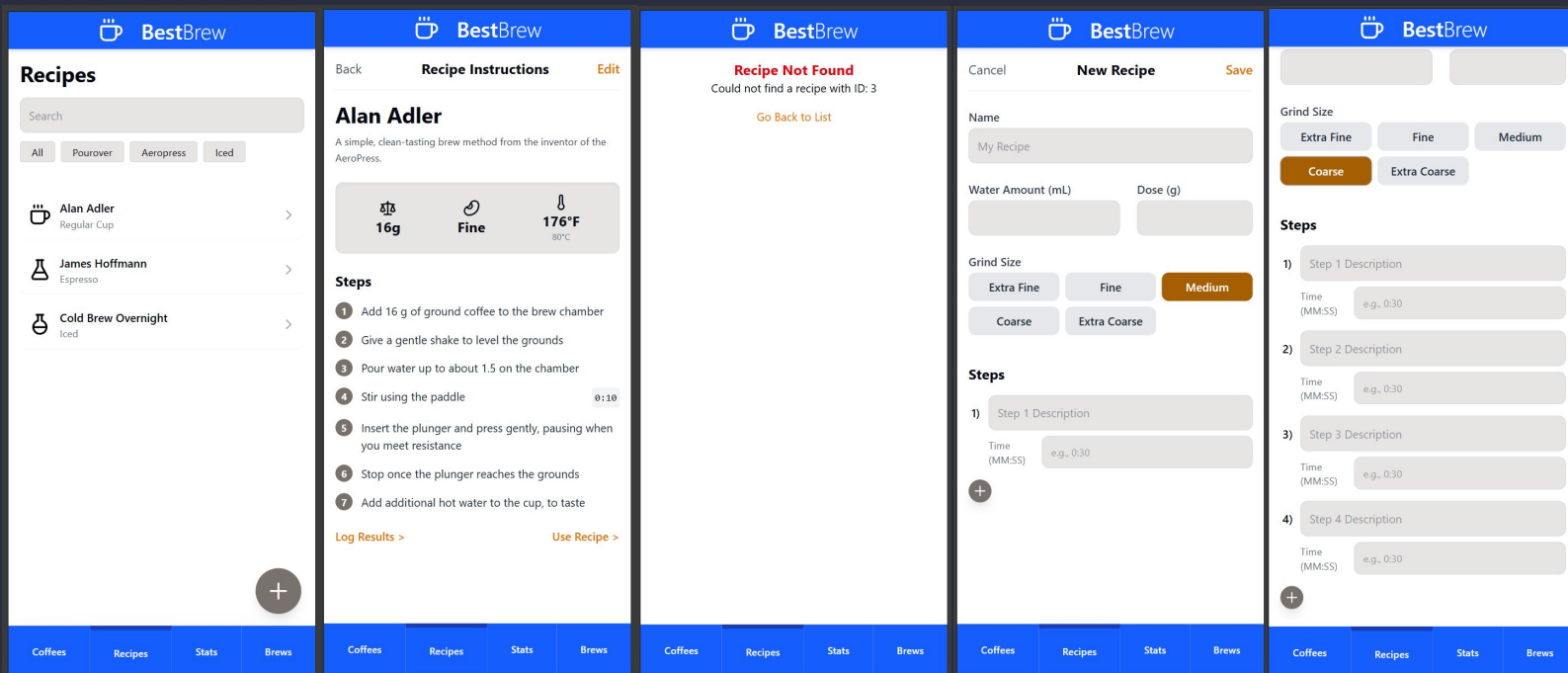
Data Logging

Your Coffees

Finca El Diviso	Worka Sakaro
Onyx Lab	Madcap

Finca El Diviso
Onyx Lab
Origin: **Colombia**
Variety: **Pink Bourbon**
Roasted: **10/05/2025**

Edit   Delete

tomboya AA

Tandem

Coffees   Recipes   Stats   Brews

Cancel   **New Coffee Bag**

Name

Roaster

Process

Weight

0   grams

Variety

Select variety

Origin

Select Origin

Roast Date

Select date

Submit

Coffees   Recipes   Stats   Brews

Notes

Select notes   +

Clarity   5

Acidity   5

Sweetness   5

Body   5

Bitterness   5

Overall   5

Submit

Coffees   Recipes   Stats   Brews

Login

BestBrew

Login

Email

Password

Login

Need an account? **Sign up**

BestBrew

Signup

Email

Password

Confirm Password

Create Account

Already a user? **Login**

React Demo

- Future Plans:
  - Adding more functionality
  - Finalize colors and design
  - Improving ease of use and accessibility