Wesley Leon, Lucas Zheng, Sopiha Chi, Eviss Wu
Yakubian Matrix
Design Document
Target Ship Date: 2026-01-16

OVERVIEW:

Our project will be a simple farming incremental simulation game using text, including farming tools and crops.

FEF: Tailwind:

Chosen for the team ’ s prior experience and ease of use. We plan on using features like its transitions & animation, boxes, borders, and hover/focus/active state.

Javascript:

Used to build cache

COMPONENT MAP:

app.py:
Configures the Flask app and handles routing, sessions, and user actions.

build_db.py:
Builds the database and populates it with crop and tool data from APIs.

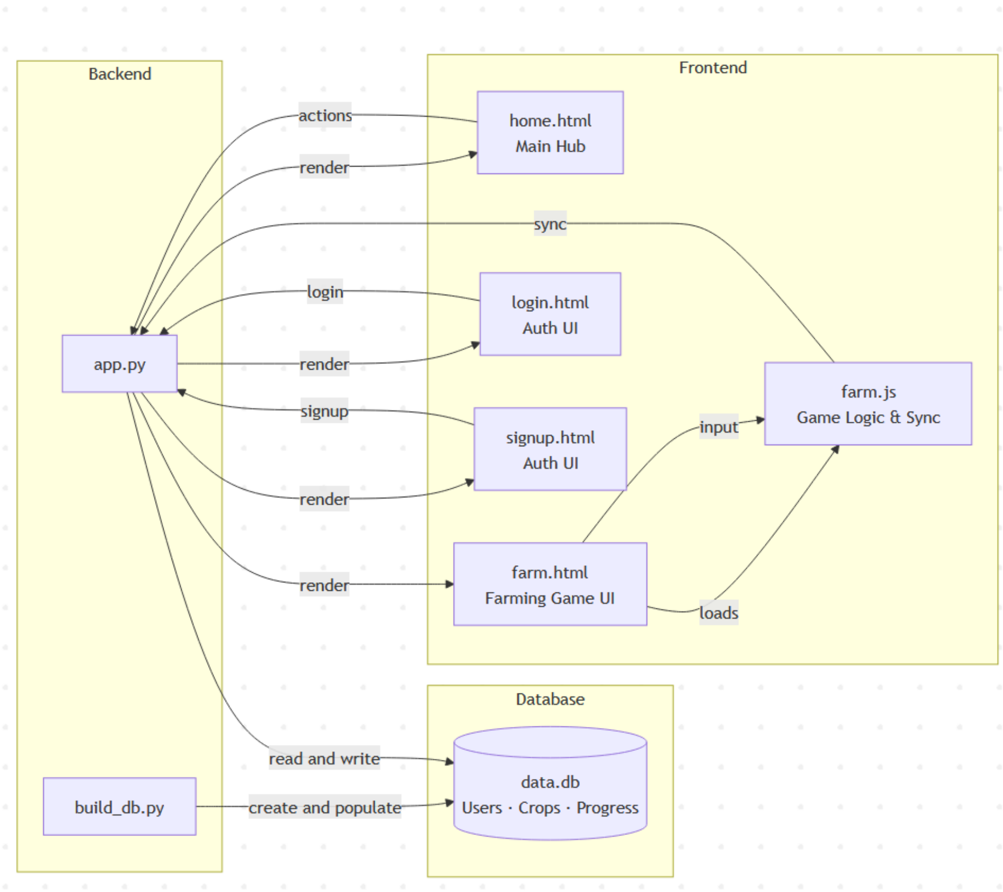
data.db:
SQLite database storing users, crops, tools, and progress.

home.html:
Main hub page linking to the farming game and displaying user progress.

farm.html:
The text-based incremental farming game interface.

login.html & signup.html:
User authentication pages.

farm.js:
Handles game logic and syncing to database.



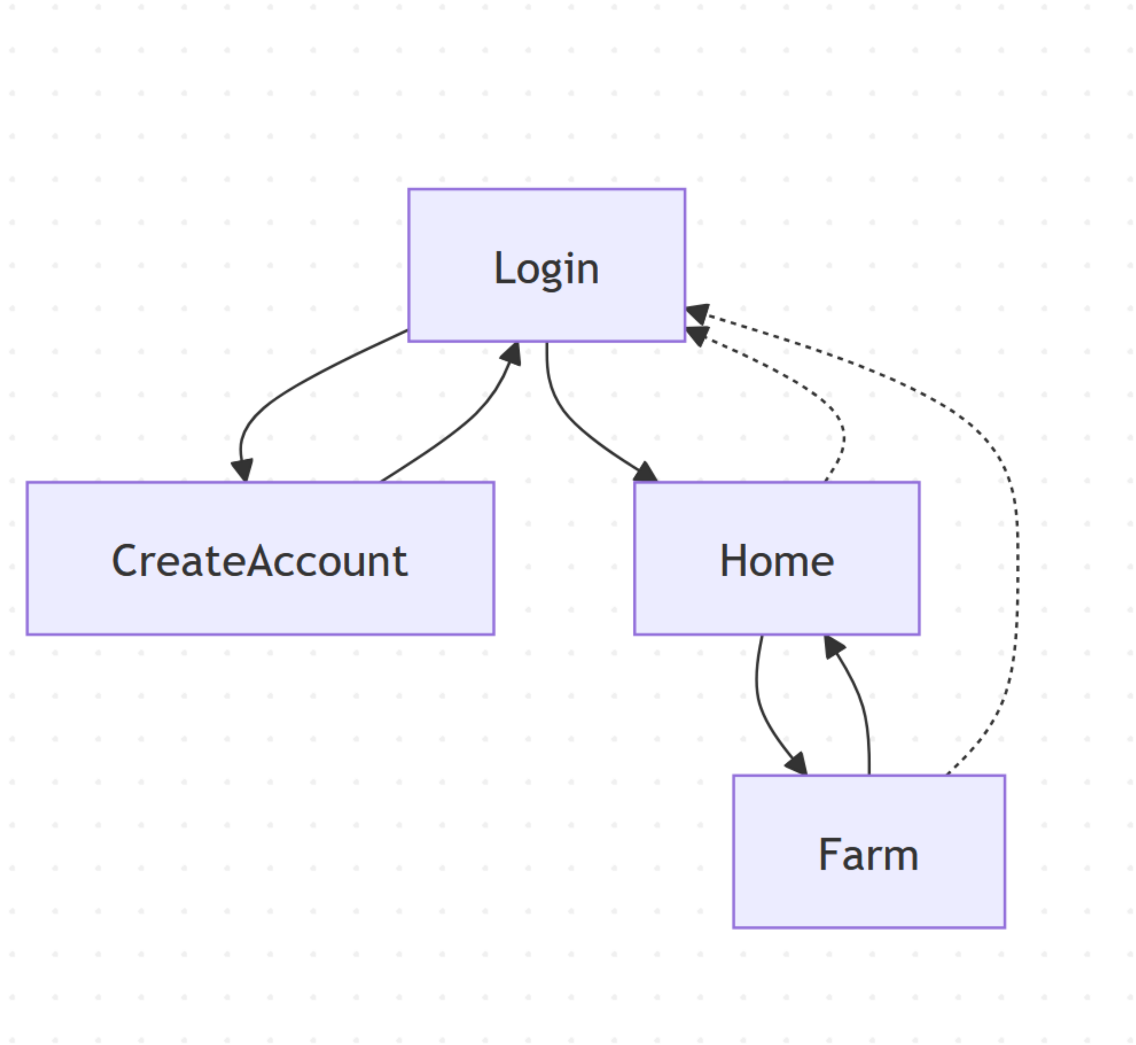
SITE MAP:

Login:
All pages should be able to go to Login. Login sends the user back to the previous page.

Create Account:
For creating an account if you don’t already have one. Only routes from Login.

Home:
Main page of the site. Routes to the farming game.

Farm:
The text-based incremental farming game. Routes back to Home.



DATABASE MAP:

Users:

id: unique user identifier

username: unique username

password: user password

Crops:

id: unique crop identifier

name: crop name

cost: cost to purchase the crop

click_value: value gained per click

growth: growth rate or growth requirement

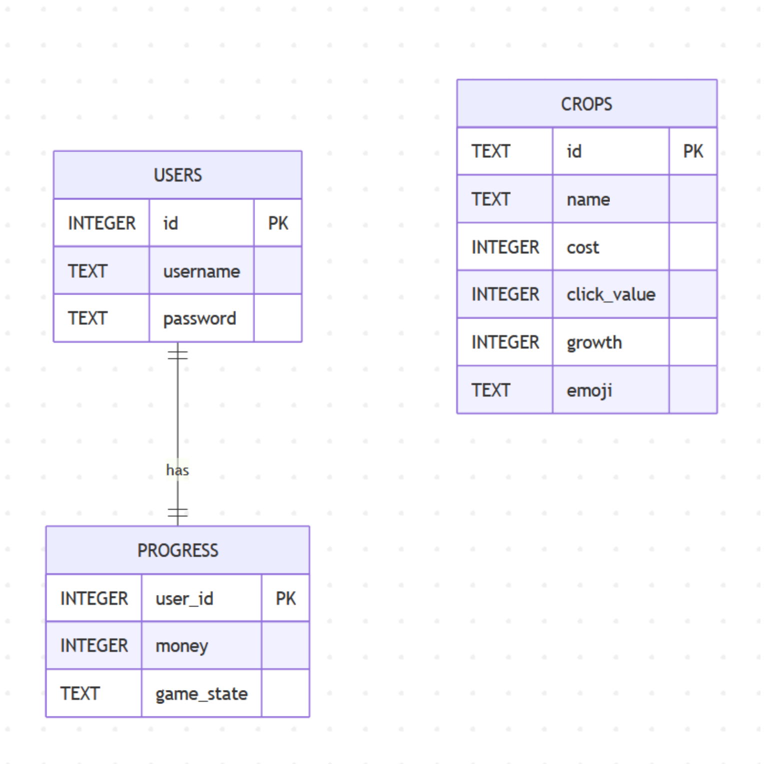
emoji: emoji representing the crop

Progress:

user_id: references the user

money: current amount of money the user has

game_state: game state data json



USERS		
INTEGER	id	PK
TEXT	username	
TEXT	password	

CROPS		
TEXT	id	PK
TEXT	name	
INTEGER	cost	
INTEGER	click_value	
INTEGER	growth	
TEXT	emoji	

PROGRESS		
INTEGER	user_id	PK
INTEGER	money	
TEXT	game_state	

Tasks:

Project Manager (Wesley):

Manages timeline and coordination.

Implements login and authentication system.

Creates Home page and user progress display.

Developer 2 (Lucas):

Implements text-based incremental farming logic.

Handles crop growth, timers, and updates.

Developer 3 (Eviss):

Handles API requests and database population.

Integrates crop and tool data into gameplay.

Developer 4 (Sophia):

Implements Tailwind CSS styling.

Improves layout, transitions, and overall UI consistency.