CamoFlask - Kendrick Liang, Derek Song, Zane Wang, Wei Wen Zhou
SoftDev1 pd8
P #02: The End
2019-01-08

Pay Day

Project Description:

Our project is a farming simulator fashioned similarly to games such as Simcity and Civilization. After creating an account, the user will be able to create their first farm. Afterward, future logins will redirect to your most recent view farm. The background and crop change may change depending on the weather and time of day. Users will be able to sell their crops to the regularly spawn NPCs for a given price. Users can also view other users' farms, add users as friends, and engage in trades with each other.

APIs used:

- aWhere: This API can be used to provide information about the various crops that the player will be growing.
- Ipapi: To get the information about location based on ip address
- Accuweather Locations API: access to geographical data of a given location (i.e. city name)
- Climacell: access to real-time weather and forecasts
- World Weather Online: access to historical weather data
- BarChat API : Get bids on crops

Front End Framework:

We are using Foundation. We chose Foundation over Bootstrap primarily because foundation's grid layout is more flexible and simplistic than bootstrap's. While bootstrap has more detailed documentation, we believe this aspect would be more harmful to us than beneficial. Since we are planning to use a lot of graphics in order to create our game, we would like to have a simple and robust FEF that gets the job done without over the top styling, but also be able to customized with a unique feel.

Features:

- Login Page: Starting page where the user enters username and password or clicks the register button to make a new username, password.
  - A new user is prompted to customize their first farm (free) based on their current location or a selected location in the world.
- View Profile:

  Allows the user to see …
    - his farms
    - crops that are grown in each farm

        - the total area of farmland owned
        - the current amount of crops owned
        - the current amount of cash the user has.
    - The user can choose to buy farmland (Cost based on the amount of land already owned).
    - Add other users as friends
    - Set their farm to be public, private, or friends view.
    - Trade with other users
- View Other Profiles:
    - Allows the user to see other users' farms and crops being grown
- Farm Layout: Shows layout of farmland after logging in. Has buttons at the bottom to designate a crop to a certain section of the land.

Database Schema:

Users

Username	Password	Cash	Friends
TEXT (PRIMARY KEY)	TEXT	FLOAT	TEXT

Farms

Owner of farm	Farm Name	Location	Area of Land	Start Time	Crops	Map	Visible
TEXT	TEXT	TEXT	INT	TEXT	TEXT	TEXT	INT - 0 (public), 1 (private), 2 (friends only)

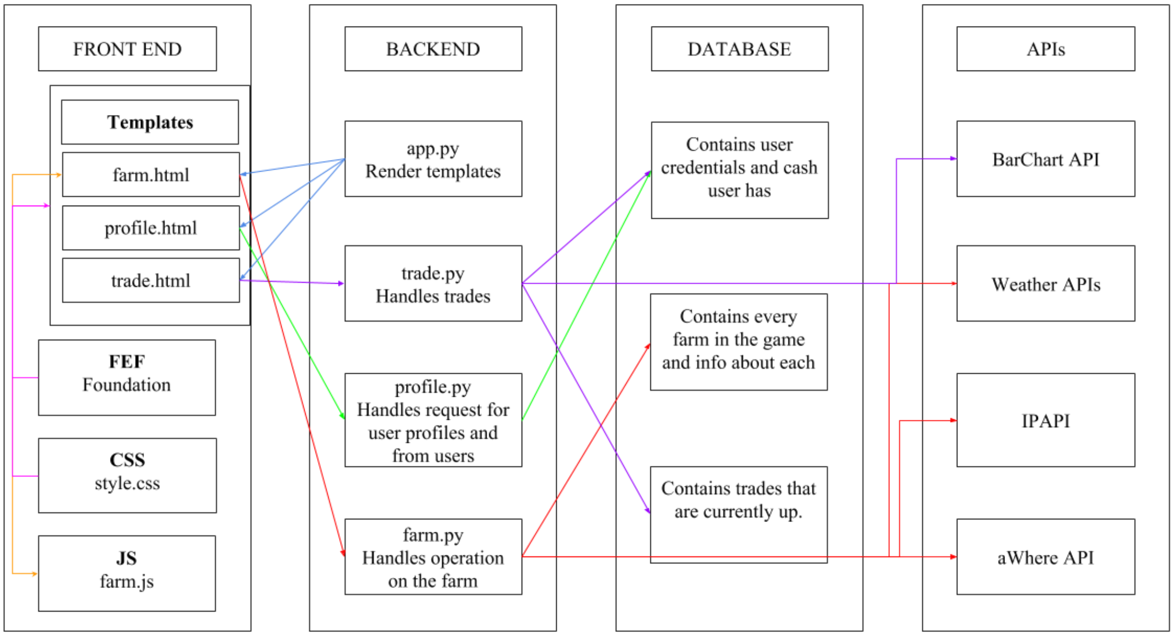
- Map : String representation containing information regarding the layout the of the farm

Trades

User	Item Being Sold	Amount Being Sold	Price per bushel
TEXT	TEXT	INT	REAL

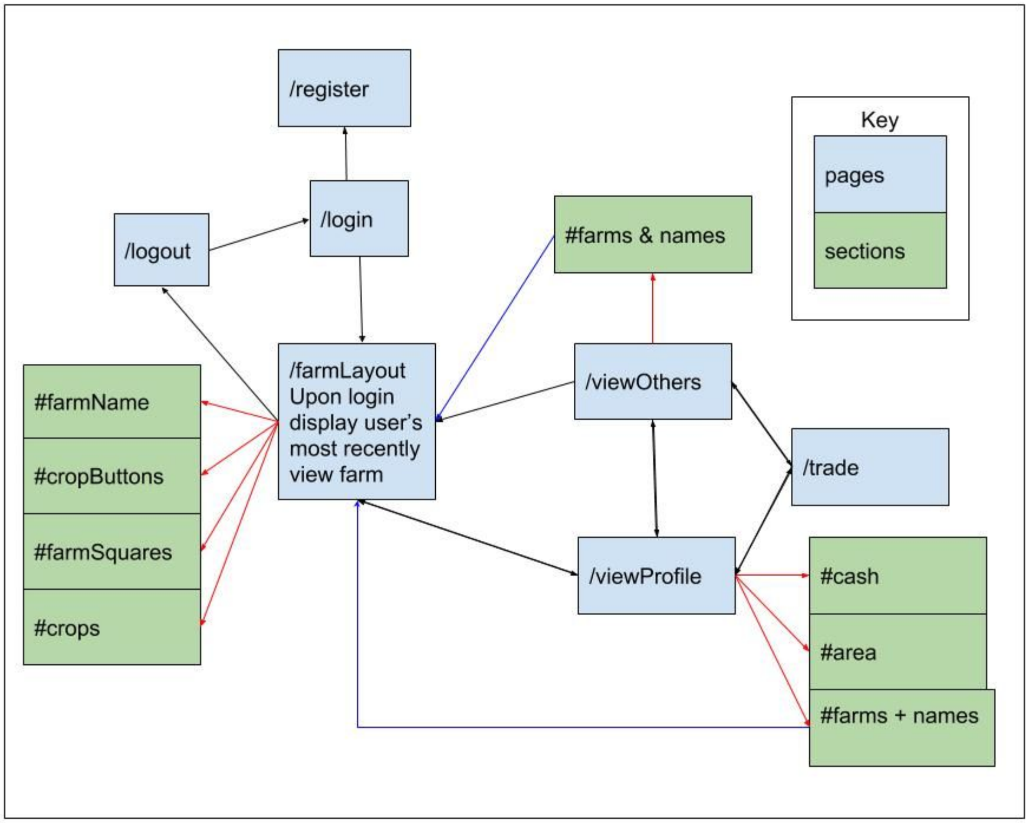
CamoFlask - Kendrick Liang, Derek Song, Zane Wang, Wei Wen Zhou
SoftDev1 pd8
P #02: The End
2019-01-08

Components Map:



FRONT END
Templates
farm.html
profile.html
trade.html
FEF
Foundation
CSS
style.css
JS
farm.js
BACKEND
app.py
Render templates
trade.py
Handles trades
profile.py
Handles request for user profiles and from users
farm.py
Handles operation on the farm
DATABASE
Contains user credentials and cash user has
Contains every farm in the game and info about each
Contains trades that are currently up.
APIs
BarChart API
Weather APIs
IPAPI
aWhere API

Site Map:



CamoFlask - Kendrick Liang, Derek Song, Zane Wang, Wei Wen Zhou
SoftDev1 pd8
P #02: The End
2019-01-08

Breakdown of Tasks:

Project Manager: Kendrick Liang
Back End DB: Creating the database and methods that can be called.
Back End API: Creating API keys, linking them to get data for function calls
Back End Critical Components: Key components of generation of time spent playing, the core functionality
Front End Framework Development consisting of CSS, Javascript, and HTML: Everyone after we are done with respective parts
Optional Features:

- Adding more farms
- Plant type differs with location

Timeline of Events:

1/10 First, we will create a profile page with login and log out, and develop .html files for profile and front end displays. (Zane)
1/10 Then, we will create a rough map in the front end framework rendered farm page. (Derek)
1/11 Then, develop the plant function. Use API, and eventually harvest. (Derek)
1/11 Implement harvesting of the soil, nutrients of the soil. (Wei Wen)
1/13 Add in weather conditions from APIs and effects on plants. (Wei Wen)
1/15 Add in viewing other user's profiles.
OPTIONAL: Add in trading/selling of crops.