PROJECT 2: Drawnet

Pd 9 | Chillin in the Front Seats: Jenny Gao, Kevin Li, William Soe, Max Zlotskiy

DESCRIPTION

Drawnet is a website where users can play a game similar to Pictionary. Users have two options: guess other users' drawings or create their own drawings. To draw, users first choose what they want to draw from a list of nouns. They are then led to a canvas - program like Paint - where they draw the object they chose and submit that drawing to be guessed by others. Drawers will receive notifications regarding other people's guesses of their drawings. For the drawer, the drawing is scored as follows: 20 - *n* (where "n" is the number of guessers who guessed the user's drawing incorrectly); after all, the quicker the drawing is guessed correctly, the better the quality of the user's drawing. If the guesser guesses a drawing correctly, the guesser gets 5 points. Thus, each user has two overall scores, as a guesser and as a drawer.

PROGRAM COMPONENTS

FRONT END (Bootstrap)

- ***stylesheet.css***
  - Additions to Bootstrap styling
- ***home.html***
  - Description of the project and option to login/create account
- ***base.html***
  - Logout/Profile header
- ***join.html***
  - Users creates account
- ***login.html***
  - Users log in
- ***chooseWord.html***
  - 5 buttons with words from a list of nouns (objects)
- ***painting.html***
  - A canvas where the user will paint their image
    - Color picker, brush thickness, and a restart drawing tool
  - A button to submit their drawing to be guessed by other users
  - Will have the word the user is painting written out in bold above the canvas as a reminder
- ***submitted.html***
  - Users are brought to this page after submitting a drawing
  - It confirms their submission and allows them to draw again

- ***gallery.html***
    - shows all of the user's drawings and scores for each drawing
- ***guess.html***
    - displays drawings by other users for this user to guess
- ***guessChoice.html***
    - Shows a text box so user can guess what the object is
- ***score.html***
    - Page that guessers are brought to upon guessing another user's drawing
    - shows score for guesser (5 points if guessed other user's drawing correctly)
    - gives user option to guess again
- ***guessed.html***
    - shows the user's guesses for drawings created by other users
- ***viewGuessed.html***
    - Shows individual drawings that user guessed
    - Contains info like user's guess if guess was incorrect
- ***profile.html***
    - shows profile picture, option to change profile picture, two overall scores (as guesser and drawer) and best/worst drawing (and their scores)
- ***notifications.html***
    - shows notifications regarding other people's guesses (incorrect and correct) and shows what time others guessed user's drawing
- ***view.html***
    - Shows individual notification/drawing, called from the notifications or gallery page
    - Displays information like notification's message, drawing's score, etc.
- ***paint.js***
    - Implements canvas with features that let the user draw with various pens and colors
    - converts the drawing into an image, which is sent to the API via flask or stored in the database

Note: all HTML files extend base.html

BACK END

- ***app.py***
    - See "Routes" section for details
    - ***isnull(usrn, pwd, conf_pwd)*:** checks to see if user filled out form incorrectly

  - ***add_session(usr, pwd):*** if the form is filled out correctly and the credentials can be authenticated in the database, creates a session
- *users.py*
  - **user_exists(username)**: returns True if this username is taken
  - **get_user_stats(username)**: returns a dictionary filled with info about user, as well as two inner dictionaries (one for best drawing and one for worst)
  - **validate_login(username, password):** checks credentials
  - **add_new_user(username, password, pfp):** adds username and password data obtained upon creation of account. PFP should be a link to an image or an empty string.
  - **get_notifications_for(username)**: returns a list of dictionaries representing the notifications for the user. They will be ordered by date, and each one will contain the keys *message*, *link*, *seen* (boolean).
  - **add_notification_for(username, message, link)**: *stores a new notification*
  - **read_notification(username, timestamp):** marks a notification as read. Requires the username of the recipient and the time the notification was sent to him/her. Both of these can be gotten from get_notification_for()
  - **add_guess(username, drawing_id, guess):** stores a guess in the database. If the guess is correct, the guesser and artist are awarded points and this method returns True. Otherwise, this method returns False. Note, you should call update_scores_for(whoever the artist is) so that their best and worst images are updated.
- *drawings.py*
  - **add_drawing(username, encoded_image, word)**: stores new drawing made by the specified user..
  - **get_image(id)**: returns a dictionary representing the image. The keys will be artist, id, image (base64), attempts (dictionary of username:guess), and solved (boolean). There is also a key called guesses, which is a list in chronological order. Every item in this list is a dictionary with the keys username, guess, and when. The correct guess will always be the last item in this list, unless the image hasn't been solved yet.
  - **get_num_guesses(id)**: returns the number of incorrect guesses for an image given its id
  - **get_images_by(username)**: *returns a list of dictionaries representing the images drawn by the user. Each dictionary will be in the format of get_image()*
  - **get_guessed_images(username):** *returns a list of dictionaries representing the images guessed by the user. Each dictionary will be in the format of get_image()*

  - **get_images_of(word)**: *similar to get_images_by(), but it searches by word rather than artist.*
  - **random_drawings(username)**: randomly generates five drawings created by others for this specified user to guess (making sure this user has not guessed this drawing yet). Returns a list of dictionaries, where each one is in the form of get_image()
  - **get_answer(id)**: returns the name of the object given an image id
- *word.txt*
  - *List of nouns (objects)*

ROUTES

- **Home Page ("/")** - *renders "home.html"*
  - Explains how the website works
- **Create Account ("/account/create")**
  - asks for username and password, will redirect to login on success
- **Login ("/account/login") -** *renders "login.html"*
  - Login or create an account
- **Guess ("/guess")**
  - This page displays images for the user to guess what others have drawn. Below each image is a textbox to submit what the user thinks the object drawn is.
- **Score ("/guess/score")** - *renders "score.html"*
  - users are directed to this route after guessing another user's drawing
  - shows guesser's score for current drawing (5 points if correct)
  - User has the option to guess another drawing, leads back to ***"/guess"*** route
- **Drawings Guessed ("/guessed")** - *renders "guessed.html"*
  - Shows drawings by other users that this user had previously guessed
  - Includes this user's response regarding what object the other user had drawn
- **View Individual Guesses ("/guessed/view")** - *renders "viewGuessed.html"*
  - On the guessed page, users see all of the drawings by other users that they have guessed. Each image is a link and brings user to this route, which shows if user guessed correctly and his/her guess if guess was incorrect
- **Choose Word ("/draw/new")**
  - This page will give the user 5 words to choose from (based on the topic they chose) to base their drawing on. They will all be nouns. Upon clicking one, the user will be brought to the drawing page: **"/draw/canvas"**

- **Canvas (“/draw/canvas”)** - *renders “painting.html”*
    - paint program for user to create drawings
- **Submit Drawing (“/draw/submit”)**
    - Users are directed to this route after clicking “Submit” on the canvas page
    - User has the option to draw another image, leads back to ***“/draw/new”*** route
- **Gallery (“/gallery”)** - *renders “gallery.html”*
    - Shows history of scores
    - Shows user’s drawings
- **View Individual Drawings (“/draw/view”)** - *renders “view.html”*
    - Shows each individual drawing
    - Also shows score, number of incorrect guesses, and correct guesser if there is one
- **Profile (“/account/profile”)** - *renders “profile.html”*
    - Displays two scores: 1) score as a guesser, 2) score as a drawer
    - Shows the user's best and worst drawings
- **Notifications (“/notifications”)** - *renders “notifications.html”*
    - Shows notifications
- **View Individual Notifications (“/notification/view”)** - *renders “view.html”*
    - Shows individual notifications when clicked
    - Shows same message, as well as score
- **Logout (“/account/logout”)** - *redirects to “/”*
    - Logs out account

DATABASE SCHEMA

TABLE: users

Column Name	**Type**	**Example**
username	STRING PRIMARY KEY	drawbot5000
password	STRING	12345
pfp	STRING	http://cats.com/selfie.jpg
guesser_score	INTEGER	10
artist_score	INTEGER	2
best_img_id	INTEGER	1
worst_img_id	INTEGER	Null

TABLE: notifications

recipient	TEXT	drawbot5000
message	TEXT	Bob figured out your drawing
link	TEXT	/draw/view?id=
seen	INTEGER (boolean)	0
timestamp	TEXT (time)	2018-01-24 23:59

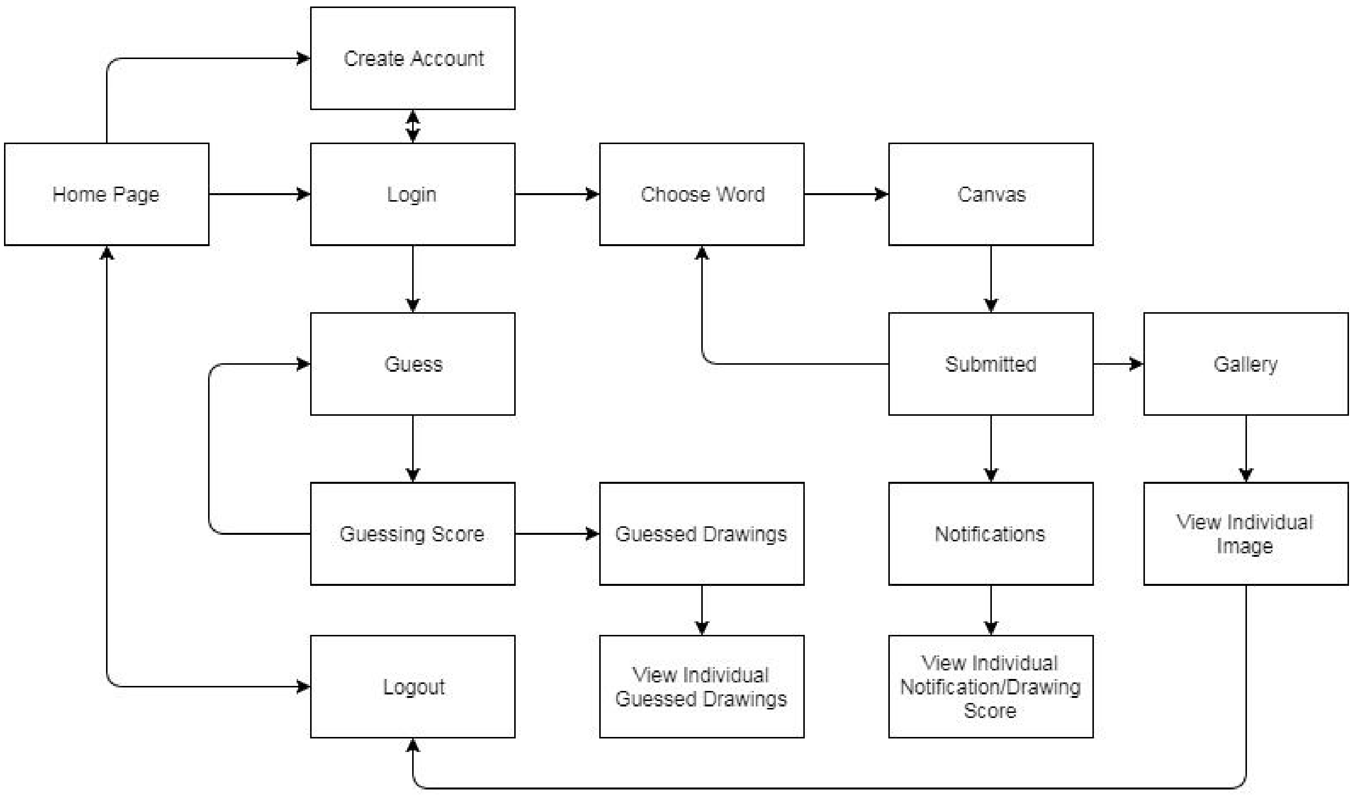
TABLE: drawings

id	INTEGER PRIMARY KEY	1
username (artist)	TEXT	drawbot5000
word	TEXT	tree
image	TEXT (base64)	[base64 encoded image]
solved	INTEGER (boolean)	0

TABLE: guesses contains a list of wrong guesses

username (who guessed)	TEXT	Bender
drawing_id	INTEGER	9001
guess (what they guessed)	TEXT	bird
datetime	TEXT (time string)	2018-01-24 23:59

SITEMAP



Note: *base.html* will have a navbar that allows users to create account, log in/log out, and view profile/gallery/guesses/notifications (so user can access these from any page)

TASK ASSIGNMENT

Project Manager: Jenny Gao
Bootstrap: Kevin Li
Canvas: Kevin Li, Max Zlotskiy
Database: Max Zlotskiy, William Soe
Flask/HTML (besides canvas)**:** Jenny Gao

Specific Tasks:

- Bootstrap - Kevin
- Canvas (HTML/Javascript) - Kevin/Max
- Playing/scoring system, account info (database) - William/Max
- HTML files/Flask app (gallery, guessed images, profile, etc.) - Jenny