

# ToDoGotech*i*

A Productivity App by Vince Casmirri, Ben Mulholland, Zach Yingling, and Marena Ferrel

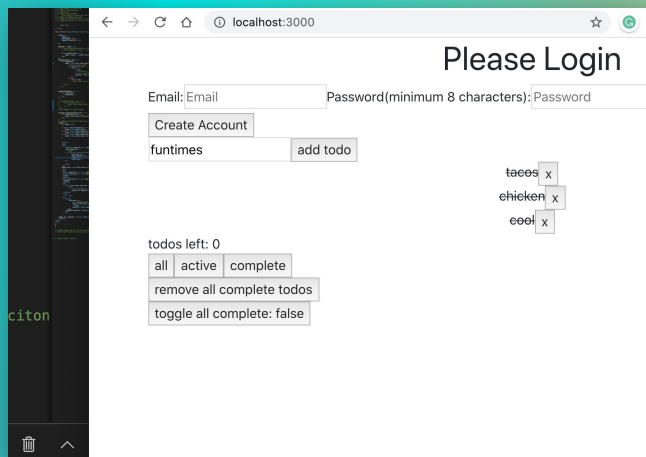


# Concept Overview

- A productivity app that stores to-dos while linking your accomplishments to the happiness of a virtual pet.
- The pet gets less and less happy over time.
- Accomplishing to-dos earns energy points that enable you to play with your pet and restore its happiness.
- ToDoGotchi aims to:
  - Harness the user's emotional attachment to the pet to improve motivation.
  - Make gamifying productivity a more gratifying experience.
  - Improve the user's time-management skills.

# Initial UI

First draft of the login screen and to-do list

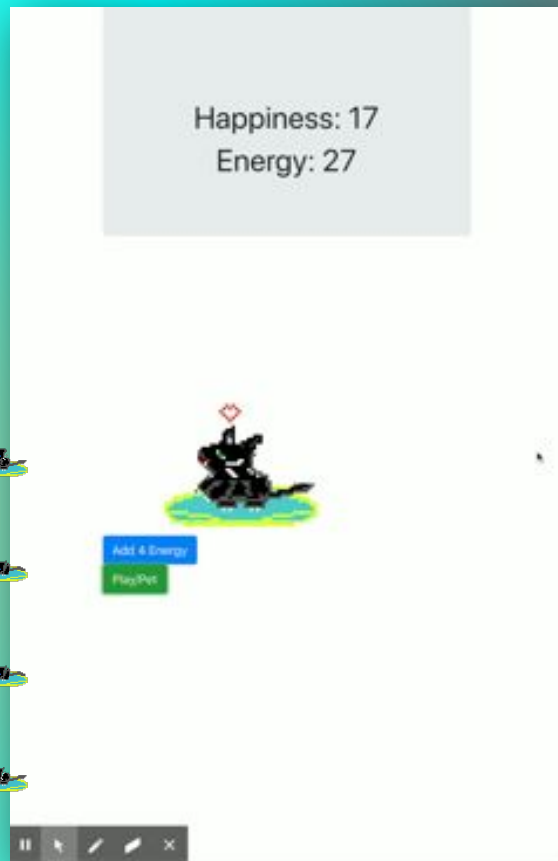


Our theme was inspired by several sources including the Tamagotchi Encore look and Lisa Frank.



First iteration of the pet page

Ben designed all of the pets and their movements



# Account Creation

- Express-Sessions
  - Stores login info
  - Login info can be access in request parameter of each route
- Email Validator
  - Validates emails on creation
- Bcrypt
  - Encrypts account passwords on creation
  - Tests if plain text matches hashed pass in database on login

# Database

- Mongo database with 3 collections
  - Users - contains login info, todolists, and a unique pet, is the parent of the following 2 collections
  - Todos - contains todo text, boolean for completion, last update timestamp
  - Pets - energy level, mood status
- When new account is created, new pet is created - object ID attributed to current user
- New todos are attributed to a specific user by the unique objectID
- Current pet state is also sent to the database where it is stored and continuously updated, the contentment of the pet depletes over time

# Minigame Probs

- Decomposition and abstraction of minigame
- Used skifree.js; Canvas, HTML5 boilerplate; piskelapp for pngs and gifs.
- Time spent animating monster, cat jumps, and crashes.



# The TodoList Component

- Users can:
  - Create new todos
  - Check off todos
  - Delete todos
  - Toggle between active and completed todos
  - See number of remaining todos

Please Login

Email:  Password(minimum 8 characters):

Create Account

funtimes  add todo

todos left: 0

all  active  complete

remove all complete todos

toggle all complete: false

taees ☐

ehicken ☐

eeol ☐

An early iteration of the component.



# Making Our Pet Sad

- Interval functions & API calls to the database
  - track time
  - decrement happiness every 2 hours
- When user logs out, database stores timestamp
- When user logs in, back-end logic calculates decremented happiness
- Additional setTimeout function accounts for elapsed time <2 hours





# Making Our Pet Happy

- Complete a todo to gain 2 energy (API call to database)
- Spend 1 energy to pet, increasing happiness
- Full happiness - user gets access to the minigame
- Interval automatically checks happiness/energy
  - Updates indicators
  - Updates animation state



# Technologies Used

- React.js
- MongoDB
- Mongoose
- Express-Sessions
- GET/POST Routing
- HTML Canvas
- SkiFree.js
- Email Validator
- User-Authentication
- Encryption
- AJAX/jQuery
- Piskel

# Future Iterations

- Multiple pets to choose from
- Text editor - ability to set word count goals
- Sound effects
- More minigames
- Reorganize components to rely less on database calls
- Use Firebase instead of Mongo