

Night Owl Developers Documentation

Sprint 1: September 27, 2021 - October 8, 2021

Overview

- Project goal: To build a web application that functions as an “Airbnb for pets”, with features that facilitate interactions between pet owners and pet service providers.
- Application features for this sprint:
 - Frontend: Pages for account signup, login, account management, search, and features list.
 - Backend: DB tables setup, APIs for user accounts and services, authentication logic, user session logic
- Team organization:
 - Frontend: Frank, Ivan, Jonathan, Meng
 - Backend: Gary, Jesse, Tyler

Frontend:

The frontend is built with React. Libraries used include Bootstrap. Below are the frontend features prioritized in Sprint 1.

Registration page: Users can register an account on the website by providing a username, password, email, and location.

Login page: Users can login to the website by providing their username and password.

Account page: After users are logged in, they can access their accounts page. On this page they can see options to edit their account information such as password, email, and location.

Search page: Users can search for service providers by using location and date.

Backend:

The backend server logic is built with Express and Node. Other Node modules used include Passport.

PostgreSQL is used as our database management system. Currently there are two tables:

users

uid	integer
username	varchar(150)
password	varchar(150)
email	varchar(250)
location	varchar(150)

services

id	integer
content	varchar(250)
title	varchar(250)
image	varchar(150)

DB API

Server location: <http://localhost:3001>

API Method	Description	Request Body Format
POST /api/auth/login	Login as a user using the login credentials.	{ username: string, password: string }
POST /api/auth/logout	Log out of the user's account.	
GET /api/user	Gets all rows of users from the database.	
POST /api/user	Create and insert a new user into the database.	
GET /api/user/:uid	Gets the user with the given uid.	
PUT	Edit the user's password,	

/api/user/:uid	email, or location fields. Username cannot be changed. At least one of these fields must be in the request body.	
GET /api/services	Gets all rows of services from the database.	
POST /api/services	Create and insert a new service into the database.	{ title: string, content: string }
PUT /api/service	Edit a service in the database.	{ title: string, content: string }
DELETE /api/services/:id	Delete a service with the matching id in the database.	