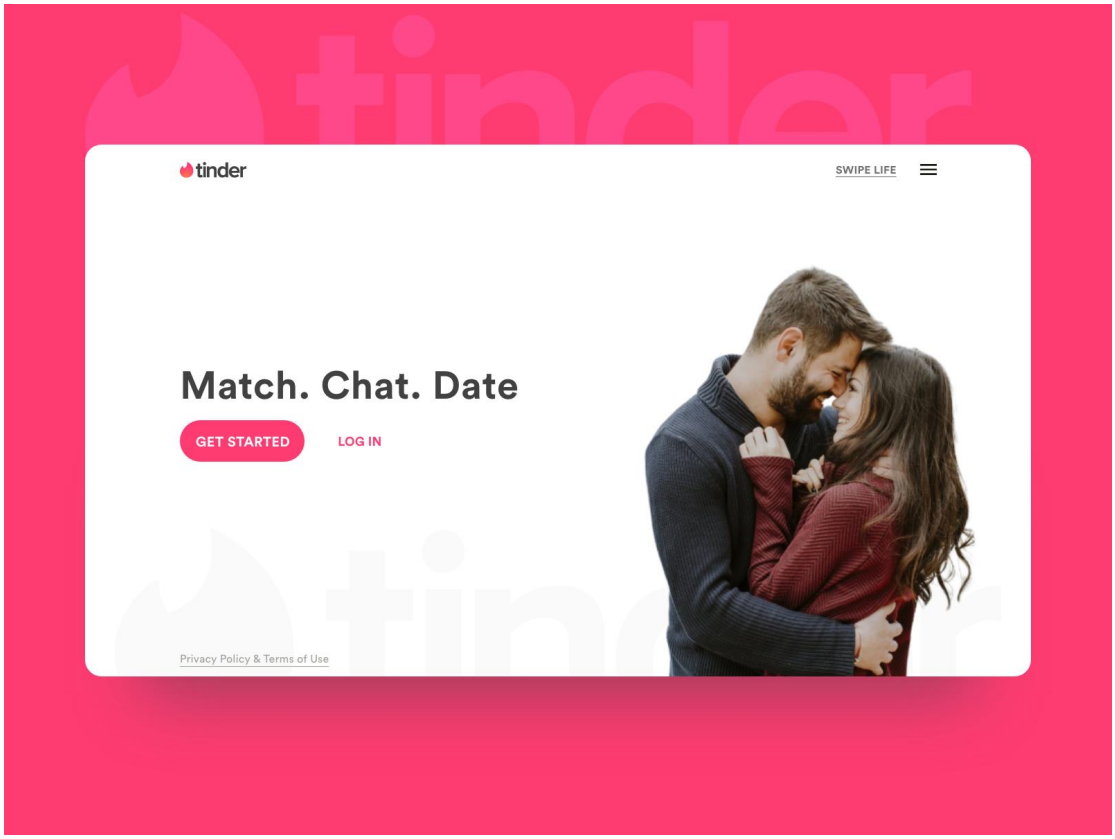
Team
Team members: Li Shenyu (team leader), Qin Letian (team member)
GitHub Classroom Repo: [GitHub Classroom link] – Shenyu Li will host the project repository.

Project Objectives
Develop a Pet Tinder website that allows pet owners to create profiles for their pets, including adding photos and liking other pets' profiles. The website will also incorporate a Maps API, allowing local pet owners to list their pets for public activities, such as group walks, thereby promoting social interaction between pets and their owners.

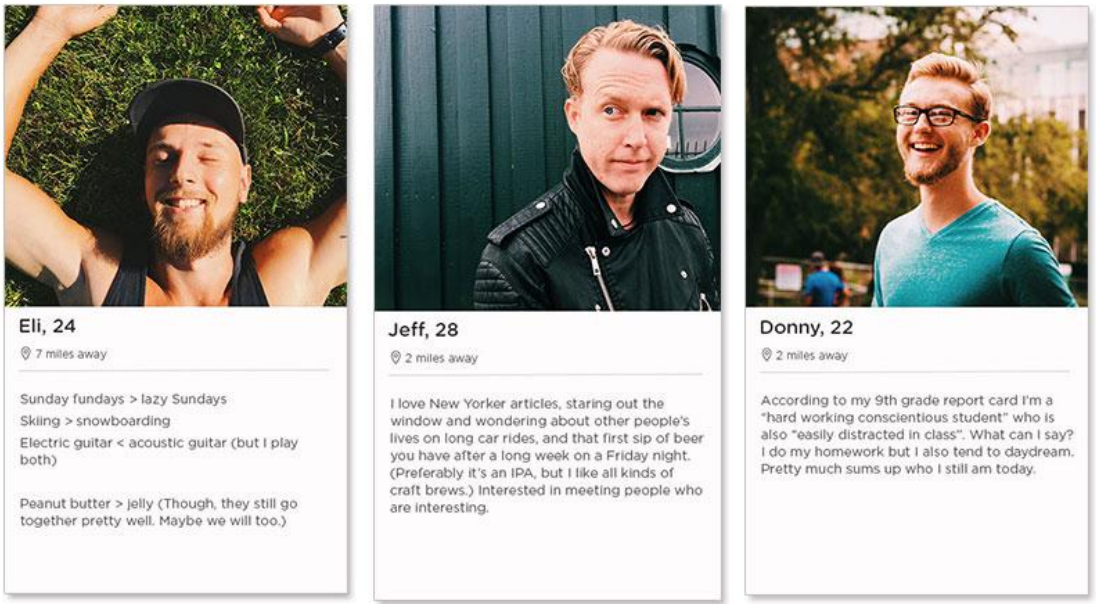
user stories
Pet Owner: As a pet owner, I can
create a profile for my pet,
add photos,
find other pets nearby to socialize with

Community Member: As a community member I would like to
explore pets in my area--profile
organize or join group walks.

User interface design
The user interface will be simple and engaging, focusing on ease of use. We will use Figma to design the interface, including:

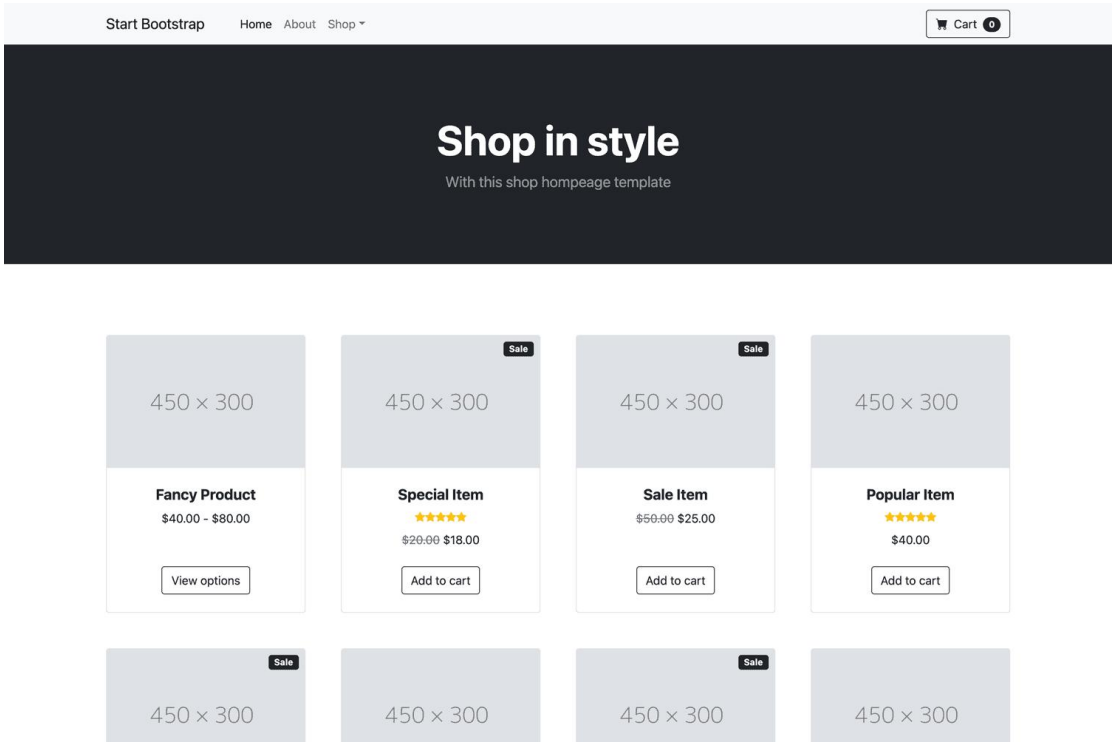


A landing page with a clear call-to-action (e.g. “Join Now”)

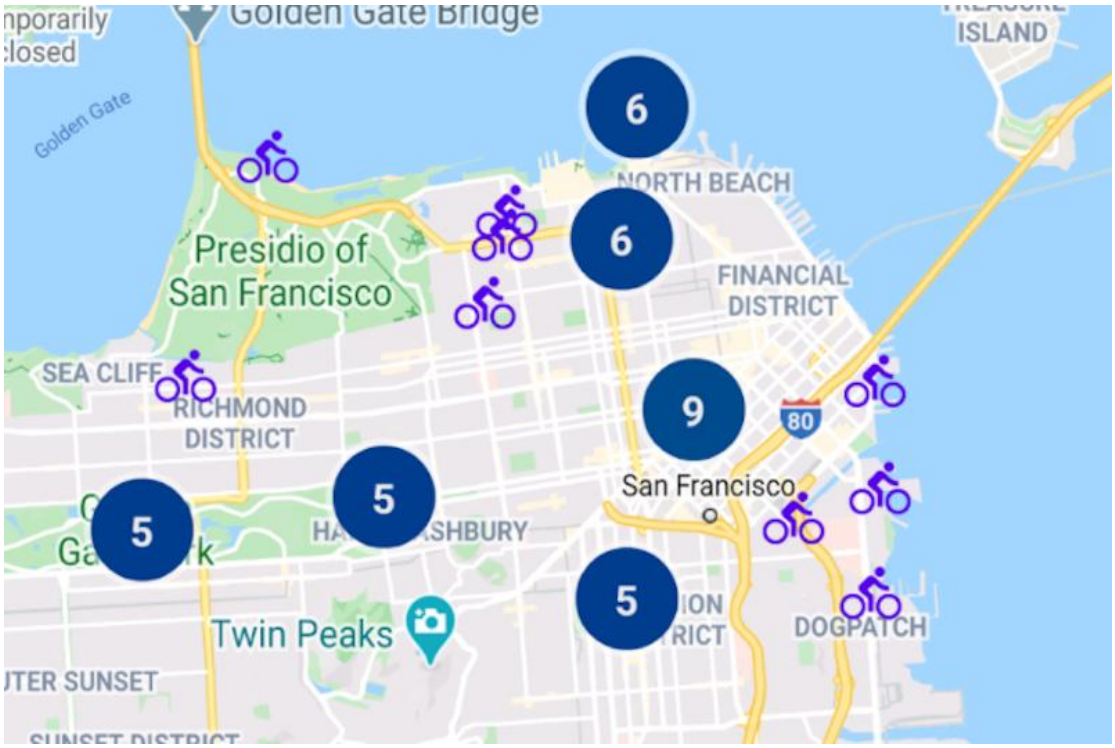


Pet's profile creation page, including name, age, breed, and photo upload fields

Main feed showing pet profiles with similar buttons and connection options



Map interface for viewing and joining local pet events



Architecture and technical requirements

Python & Django

Back-end development: Utilize Django's back-end logic and database interaction capabilities to handle back-end services such as user authentication, data processing, and API development.

Database interaction: Django ORM interacts with MySQL database to manage user information, pet information, like records and other data.

Angular

Front-end development: Creating user interfaces using Angular. Angular's two-way data binding and modular development methods will be used to develop user profile pages, pet browsing interfaces and activity map interfaces.

MySQL

Data storage: Use MySQL database to store user data, pet data, like information and activity information.

Other technologies

Authentication and security: JWT (JSON Web Tokens) can be used for authentication to ensure the security of user sessions.

Front-end and back-end communication: REST API design pattern, use Django REST framework to create API, realize front-end and back-end separation, and improve the maintainability and scalability of the system.

Map API integration: Use Google Maps API or OpenStreetMap API to implement map functions, display activity locations, and support users to discover and participate in nearby activities.

Other requirements

Acknowledgments: We will cite all sources used for development, including but not limited to MDN, textbooks, and reputable open source projects.

Reproducibility: Detailed instructions for setting up the development environment and running the application will be provided in the README.md.

Deployment: The project will initially be set up against a local development server. If

we decide to deploy it publicly, we will include a comprehensive privacy policy consistent with Berkeley guidance.
project management
Work Plan: The project management plan will detail roles, responsibilities, milestones, and timelines. This will be included in the team leader's repository as a separate Markdown file in the proposal subdirectory.
Milestones: Includes initial setup, feature development phases (e.g. authentication, profile management, map integration), testing and final deployment.

Roles and Responsibilities:

Shenyu Li: project management. Front-end development, UI design, and map API integration.
Letian Qin: back-end development, database management