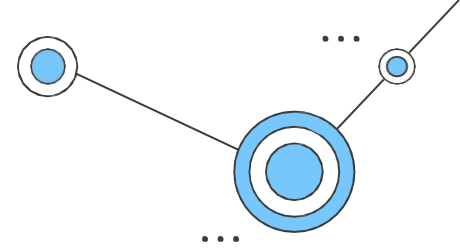


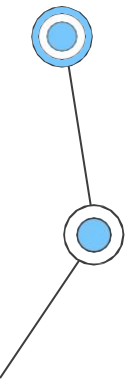
Happy Paw

Snicker Squad
2023-2024

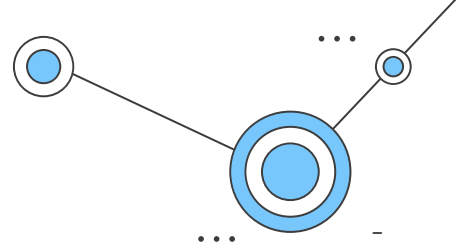
Разпределение на роли



Роля	Студент
Developer 1	Слави Манолов
Developer 2	Мариета Атанасова
QA	Димитър Казаков
Project manager	Кръстьо Танев
Product owner	Диляна Митева



Съдържание



01

Идея на проекта

Основна идея на проекта
Разпределение на екипите

...

02

Технологичен стек

Езици за програмиране и технологии
използвани за разработка на проекта

...

03

Архитектура на проекта

Основна архитектура на проекта
Комуникация между сървър и клиент

...

04

Функционалности и тестване

С какви функционалности разполага проекта
Начини за тестване на функционалностите

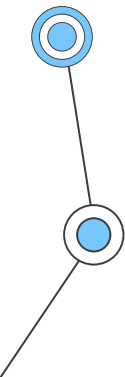
...

05

Бизнес модел

Какъв е бизнес модела на проекта
Как ще печели софтуера и финансиране

...





01

Идея на Проекта





Роли и техните дейности

Разработчикът (DEV 1) ще бъде отговорен за писането на кода на приложението. Той/тя може да използва различни езици за програмиране в зависимост от нуждите на проекта.

Разработчикът (DEV 2) ще бъде отговорен за проектирането на архитектурата на приложението. Това включва определяне на това как различните части на приложението се свързват и взаимодействат помежду си.

QA (Quality Assurance) специалистът ще бъде отговорен за тестване на приложението, за да се увери, че всички функции работят правилно. Това включва създаване на тестови сценарии, изпълнение на тестове и докладване за грешки.

Проектният мениджър (PM) ще бъде отговорен за общата координация на проекта. Това включва планиране на задачите, управление на ресурсите, контрол на качеството и сроковете, както и комуникация със стейкхолдерите.

Product Owner (PO) ще бъде отговорен за определяне на бизнес модела на приложението. Това може да включва определяне на начини за генериране на приходи, като например реклами, дарения или платени функции.

...



Идея и изследване

01

Идея

Целта на проекта е да създадете уеб приложение, което помага на животни в нужда. Приложението ще предоставя платформа за репортиране на изгубени и бездомни животни, даване на съвети за помощ на животни и предоставяне на списък за осиновяване на животни.

02

Изследване

Това включва разбиране на текущите нужди и проблеми, свързани с животните в нужда. Можете да проведете анкети, интервюта или да използвате данни от вече съществуващи източници, за да разберете какви са най-големите предизвикателства и как вашето приложение може да помогне.





03

Планиране

Създайте подробен план за разработването на приложението. Това включва определяне на функциите, които трябва да има приложението, избор на технологии за разработка, определяне на срокове и задачи, и разпределение на ресурсите.

04


Прототипиране и тестване

Създайте прототип на приложението и го тествайте с потребители, за да получите обратна връзка. Това ще ви помогне да разберете какво работи добре и какво трябва да бъде подобро преди пускането на приложението.

05

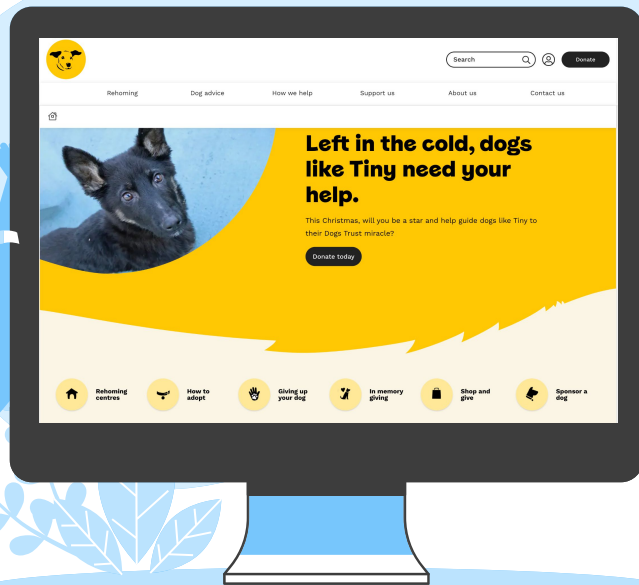
Пускане

След като приложението е готово и тествано, го пуснете в употреба. След това продължете да го поддържате и подобрявате, като реагирате на обратната връзка от потребителите и правите необходимите актуализации.

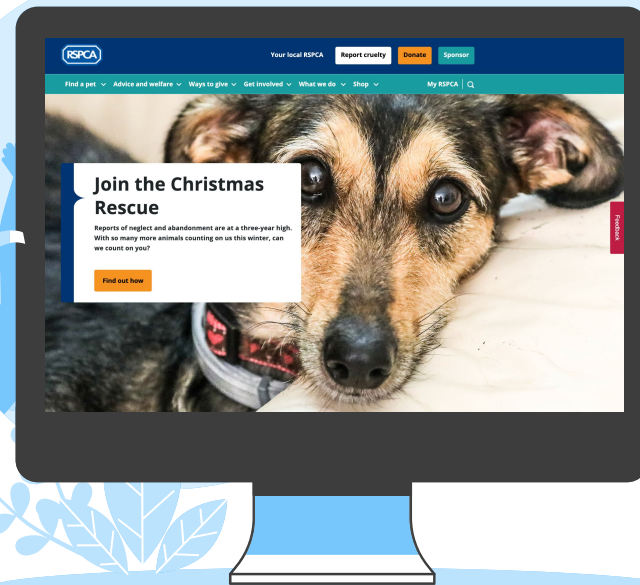


Примерни успешни проекти

Dog Trust



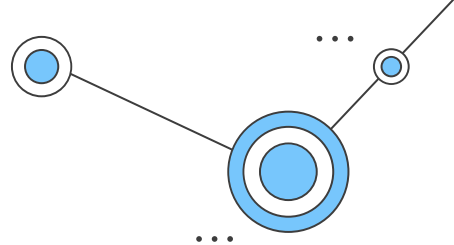
RCPSPA



02

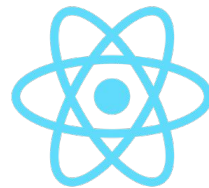
Технологичен Стак

MERN

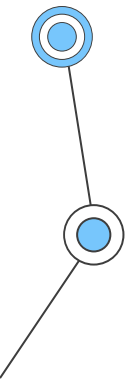


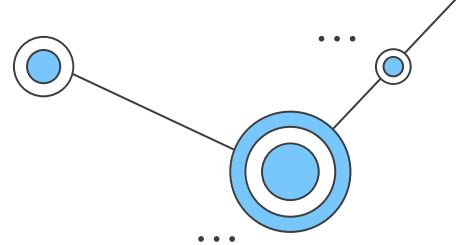
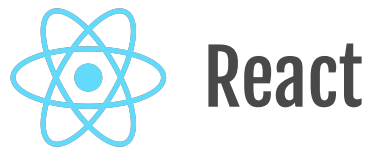
MERN е JavaScript Stack, който се използва за по-лесно и по-бързо внедряване и разработка на уеб приложения. Съставен е от MongoDB, Express, React, Node.

- **MongoDB:** нерелационна база данни
- **Express:** Node.js уеб рамка
- **React:** JavaScript Frontend библиотека
- **Node:** уеб сървър на JavaScript



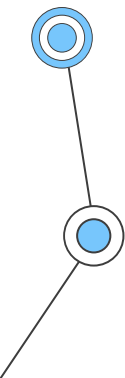
express



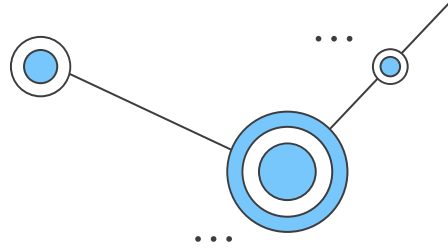


React е JavaScript библиотека за изграждане на потребителски интерфейси. Чрез него могат да се създават преизползваеми компоненти, които се актуализират в отговор на промени в данните. Той използва и виртуален DOM за оптимизиране на производителността. Заедно с React, може да се използват и следните технологии:

- React Query - за извличане, кеширане, синхронизиране и актуализиране на данните;
- Axios - за изпращане на асинхронни HTTP заявки;
- Material-UI - библиотека предоставяща готови компоненти;
- React Hook Form - прави валидирането на формуляри лесно и оптимизирано;

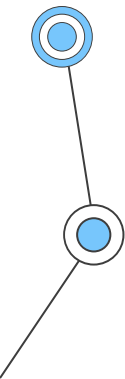


Node & Express



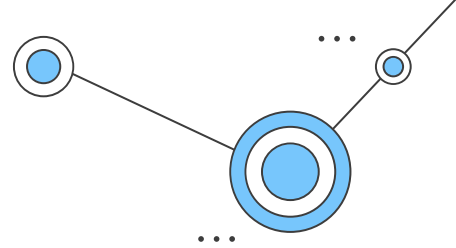
Node JS е среда за изпълнение с отворен код за изпълнение на JavaScript код извън браузър. Често се използва за изграждане на back-end услуги. Отличителна черта на Node.js е неблокиращият си режим на вход/изход.

Express е малка рамка за работа с Node JS. Използва се за управление на HTTP заявките и отговорите, дефиниране на пътища (routes), обработка на middleware, управление на сесии и още много други.





MongoDB

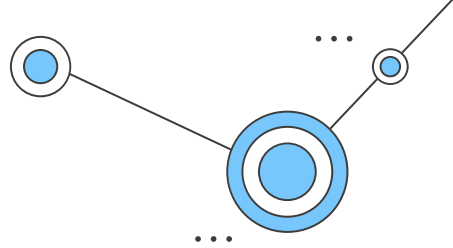


MongoDB е популярна NoSQL база данни, която съхранява данни в гъвкав формат, подобен на JSON, наречен BSON. Той е проектиран да обработва големи количества неструктурирани данни и е известен със своята мащабируемост, висока производителност и лекота на разработка. MongoDB използва документно-ориентиран модел на данни, което го прави подходящ за различни приложения, включително такива с бързо развиващи се схеми.

```
{  
  "student": {  
    "name": "John",  
    "class": "Intermediate",  
    "address": {  
      "street": "2293 Example Street",  
      "City": "Chicago",  
      "State": "IL"  
    }  
  }  
}
```



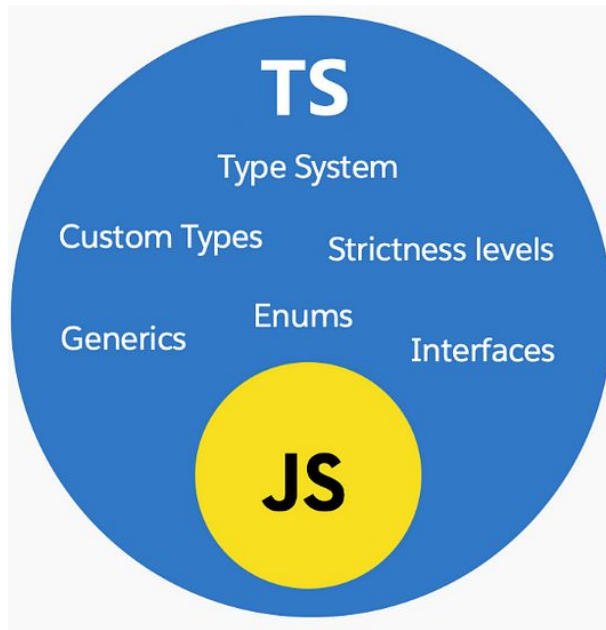
JS Javascript & Typescript TS



При работа с MERN се използва предимно JavaScript, както за разработка от страна на сървъра, така и от страна на клиента.

TypeScript е статично типизиран и обектно-ориентиран програмен език, суперсет (надмножество) на JavaScript. Може да се използва за подобряване на качеството на кода при разработката.

Докато JavaScript остава основен, TypeScript се избира в някои случаи, особено за големи проекти, за ранно улавяне на грешки и подобряване на поддръжката.

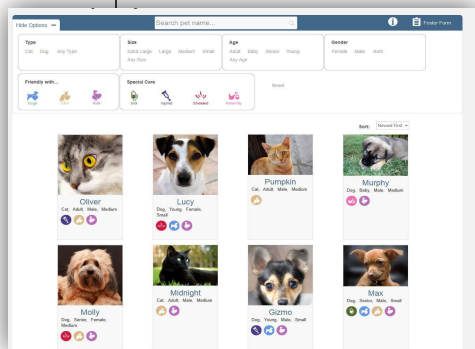


03

Архитектура на проекта

Уеб базирано приложение

- Потребителите ще имат достъп чрез интернет
- Основна цел: подпомагане на животни в нужда
- Метод на работа: форма за подаване на животни в нужда/ на сайта ще има списък с животните – къде се намират, от какво са болни, от какво се нуждаят
- Потребителите ще могат да се свързват с екипа, за да окажат помощ
- Потребителите ще могат да изпращат средства или нужни лекарства/вещи
- Потребителите ще могат да осиновяват животинките



Front end - Single page application on React

- React – базиран на концепцията на компоненти, така кода става по-структуриран
- React позволява по-ефективна актуализация на интерфейса
- За REACT има обширна документация
- Позволява се и лесно преминаване към мобилно приложение
- REACT лесно се интегрира с други библиотеки
- SINGLE PAGE APPLICATION – една html страница, която се обновява динамично – лесно добавяне на нови елементи, лесно адаптиране към други устройства

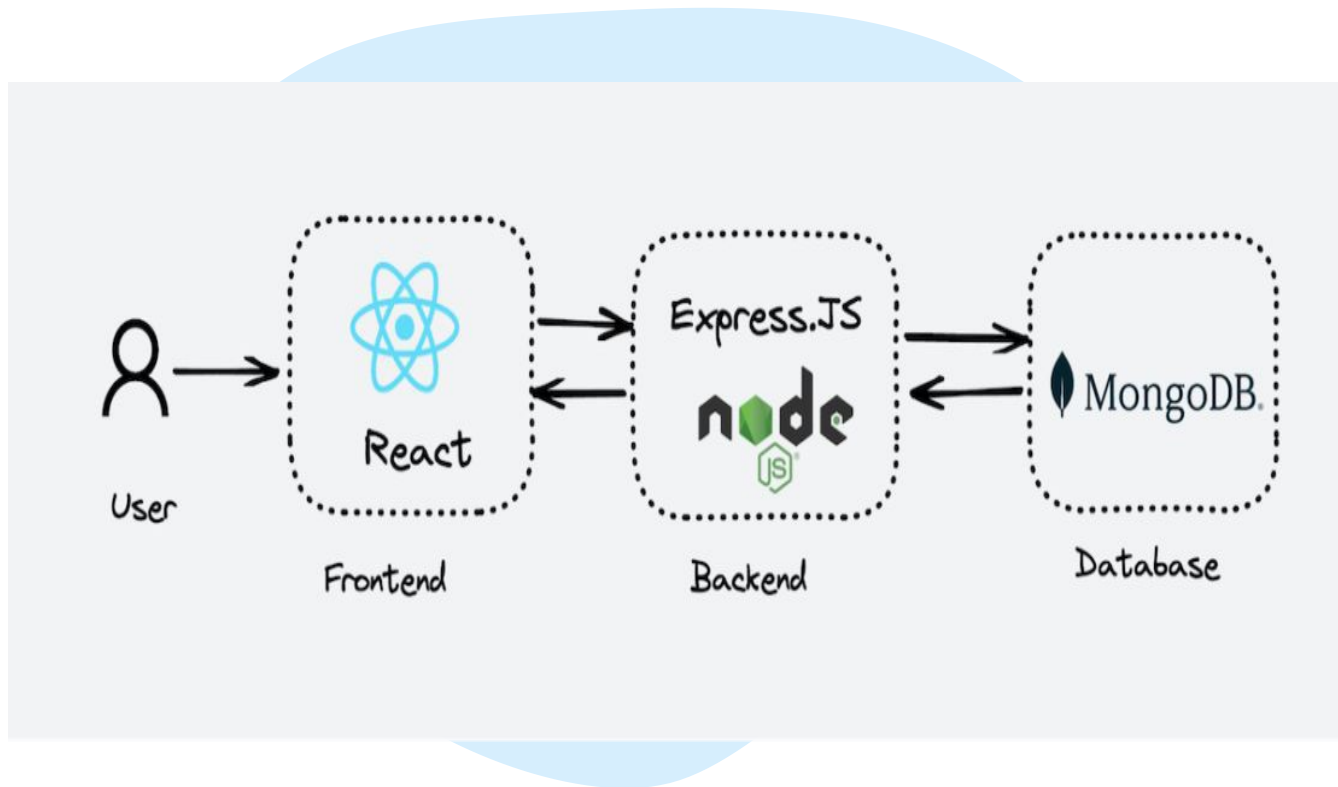
Back end - Rest API on Express and Node

- Лесна интеграция между frontend и backend
- Node.js позволява бързо изпълнение на заявките
- Лесно споделяне на код между сървъра и клиента
- Node.js, Express – гъвкавост
- Express – лесно добавяне на функционалности
- Express – лесна интеграция на бази данни

NoSQL Database - MongoDB

- Предоставя гъвкавост и мащабируемост
- Съхранява данните в JSON формат
- Богати възможности за индексирание
- Лесна интеграция с различни технологии и езици
- Поддържа разнообразни операции
- Улеснена работа с транзакции

Диаграма



04

Функционалност и Тестване

Функционалности



Намери любимец

- Карта с доклади за изчезнали животни
- Списък с приюти за животни



Форум

- Добавяне на съвети
- Дискусии за любимците



Осиновяване

- Форма за осиновяване на любимец
- Списък с любимци за осиновяване



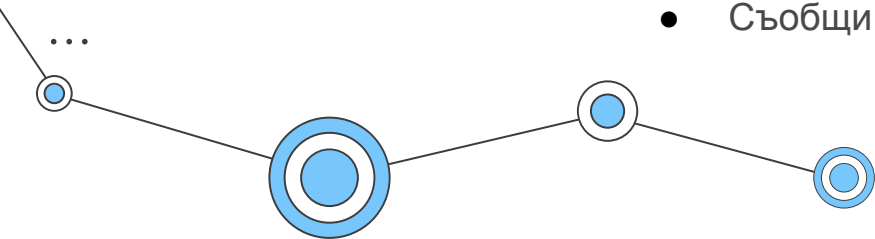
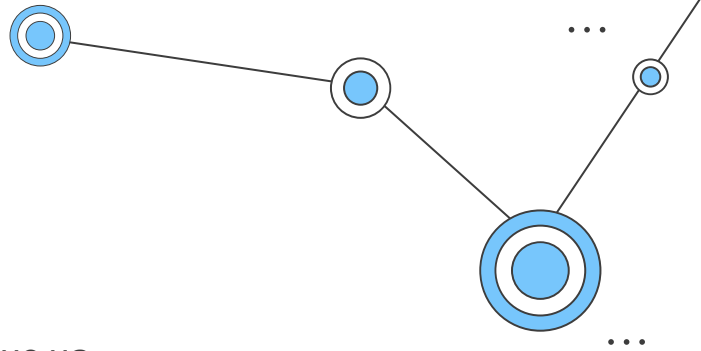
Съобщи/Докладвай

- Съобщи за изчезнало животно
- Съобщи за ранено животно
- Съобщи за бездомно животно



Дарения и помощи

- Как да помогнем ?
- Съвети за помагане
- Списък с неща от първа необходимост





Методи за тестване



01

Unit Tests

- Тестване на отделните единици в кода. Работят ли поотделно както е предвидено
- Функционално тестване - тестване на отделните функции с нагласени параметри
- Структурно тестване - тестване дали компонентите се подредени правилно при различни условия
- Jest, React Testing Library, Faker.js

02

Integration Tests

- Тестване на главни функционалности от начало до край, взаимодействие на отделни под системи, интегриране на клиент със сървър, тестване с реални данни
- **Top-down integration** тестване - тестване първо на компонентите на по-високо ниво
- Jest, React Testing Library

03

E2E Tests

- Тестване на целия поток на приложението, верифицира че всички компоненти работят добре заедно и симулира действия на истински потребител
- Описват се реални сценарии и се правят автоматизирани тестове симулиращи дейност на истински потребител
- Playwright

Детайли за функционалностите

01

Намери Любимец

Тази функционалност ще представлява карта, на която ще може да се видат доклади за изчезналите животни и къде са видени последно. Също така ще има локации на всички приюти, ветеринарни клиники и зоо магазини. Потребителят ще има възможността да прилага различни филтри.

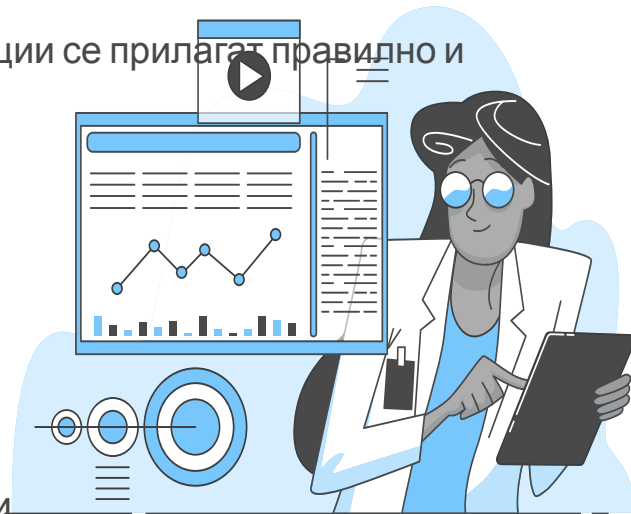
- Трябва да се тества дали правилно показваме локациите на приютите, клиниките и магазините и дали работното време и телефоните за контакти са правилни за всеки от дните
- Трябва да се тества дали потребителя вижда списъка с всички любимци и дали те са посочени на картата с правилната локация

02

Осиновяване

Тук потребителят ще може да види списък с любимци за осиновяване както и да попълва онлайн заявление за осиновяване при желание. Ще има и списък как да помогнем, ако потребителя желае да помогне но няма възможност да осинови любимец

- Трябва да се тества че списъка е актуален и информацията се взима от всички приюти
- Трябва да се тества че при осиновяване списъка се актуализира и житовното е маркирано като осиновено
- Трябва да се тества че онлайн заявленията се пращат и



Детайли за функционалностите

03

Форум

В този форум потребителите ще могат да се регистрират и да участват като дискутират различни теми за любимците си и търсят помощ. Тук ще се поставят линкове до статии и блогове които са полезни за всички. Потребителите ще могат да правят чат стаи в които да обсъждат нещата на лично и ще могат да организират събития от типа да помогнем на някой приют или да изчистим/оправим някой парк за кучета.

- Трябва да тестваме че потребителите успешно се регистрират и влизат в профилите си
- Трябва да тестваме че потребителите успешно дискутират и поставят съобщения във форумите
- Трябва да тестваме че потребителите успешно създават чат стаи и пращат съобщения там
- Трябва да тестваме че потребителите могат да създават събития и да модифицират

04

Дарения и помощ

Тук ще намерите информация как можете да помогнете с неща от първа необходимост, финансово или да допринесете с някаква дейност. Ще се поставят съвети за помагане на любимци както и събития организирани от нас за помагане на приюти и дейности като чистене и откриване на нови паркове

- Трябва да тестваме че информацията е видима от всички и при ъпдейт информацията се променя
- Трябва да тестваме че може да създаваме успешно събития и хората да коментират и участват в тях
- Трябва да тестваме че финансовите дарения



Детайли за функционалностите

05

Съобщи / Докладвай

Това е една от основните функционалности чрез които си набавяме информация. Всеки потребител ще има възможността да докладва за ред неща - като например изгубен любимец, ранено животно, улично животно без таг, нови локации като клиники, магазини и приюти, паркове и места за отдих с любимеца. Това цели чрез помощта на потребителите приложението да бъде с актуална информация. След докладване потребителя ще получи бонус точки, които ще се запазят в профила му. Тези точки той ще може да ползва в редица 300 магазини за покупка или намаление на избрани артикули

- Трябва да тестваме че потребителя може да докладва успешно и че валидации на формите работят:
 - Изгубен любимец
 - Ранено животно
 - Улично животно без таг
 - Нови локации - паркове, магазини, приюти, клиники
- Трябва да тестваме че след докладване от потребителя актуалната информацията се показва на картата (Намери Любимец)
- Трябва да тестваме че след докладване потребителя получава бонус точки и те се показват в профила му





05

Бизнес модел



Визия и Цел



Визия

01

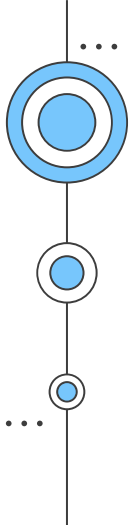
Създаване на онлайн платформа, посветена на подпомагане на бездомни и изгубени животни

Цел

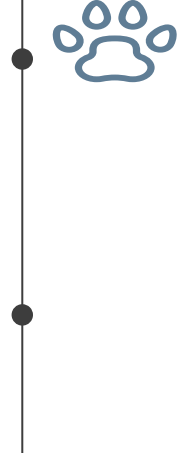
02

Улесняване на намирането на подходящи домове за бездомни животни, осигурявайки интуитивен начин за обмен на информация и подкрепа от общността

...



Бизнес Модел



Основни Услуги

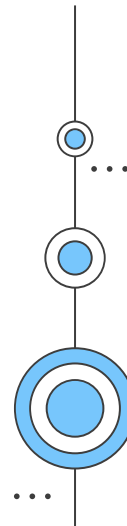
Регистрация на бездомни или изгубени животни с основна информация и снимки. Търсене и преглед на животни, които търсят дом.

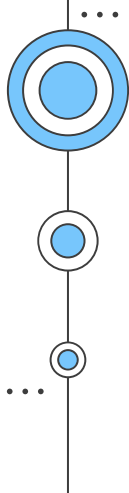
Дарителски Възможности

Възможност за потребители и организации за даряване на средства за подкрепа на благотворителни програми за животни.

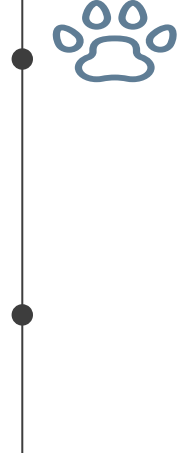
Събития и Кампании

Популяризиране на специални събития и кампании, целящи подпомагане на конкретни животни или спасителни организации.





Целева Аудитория



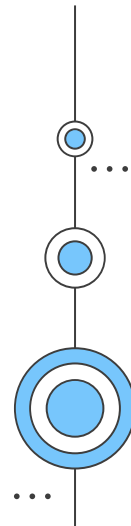
Основни Група

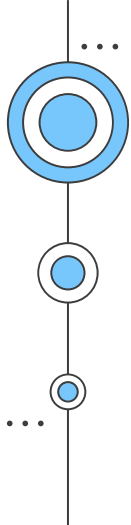
Любители на животните, които искат да помагат и да участват в подпомагането на бездомни и изгубени животни.

Организации за Защита на животните

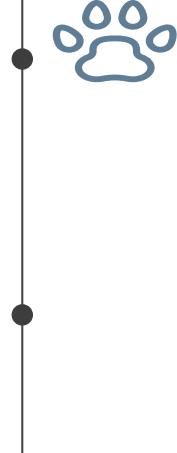
Възможности за регистрация и активно участие от страна на организации, предлагащи подслон и грижи за животни.

...





Пазарен Потенциал



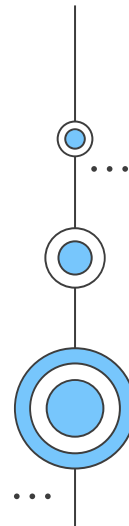
Благотворително Участие

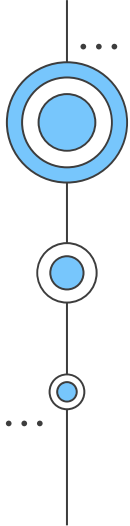
Привличане на дарители и спонсори, които се интересуват от благотворителност и защита на животни.

Сътрудничество с ветеринарни услуги

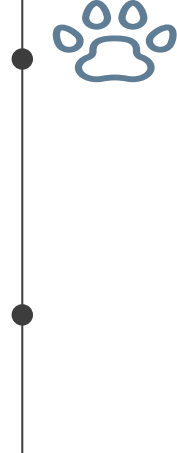
Потенциално партньорство с ветеринарни организации за предоставяне на здравни грижи за бездомните животни.

...





Печалби и Възможности за развитие



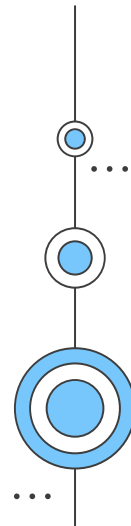
Приходи от реклами


Рекламни възможности за ветеринарни клиники, магазини за домашни любимци и други свързани бизнеси.

Допълнителни функционалности

Интеграция на допълнителни функционалности като онлайн консултации с ветеринари и образователни материали за грижи за животните.

...





Благодарим
за
вниманието!