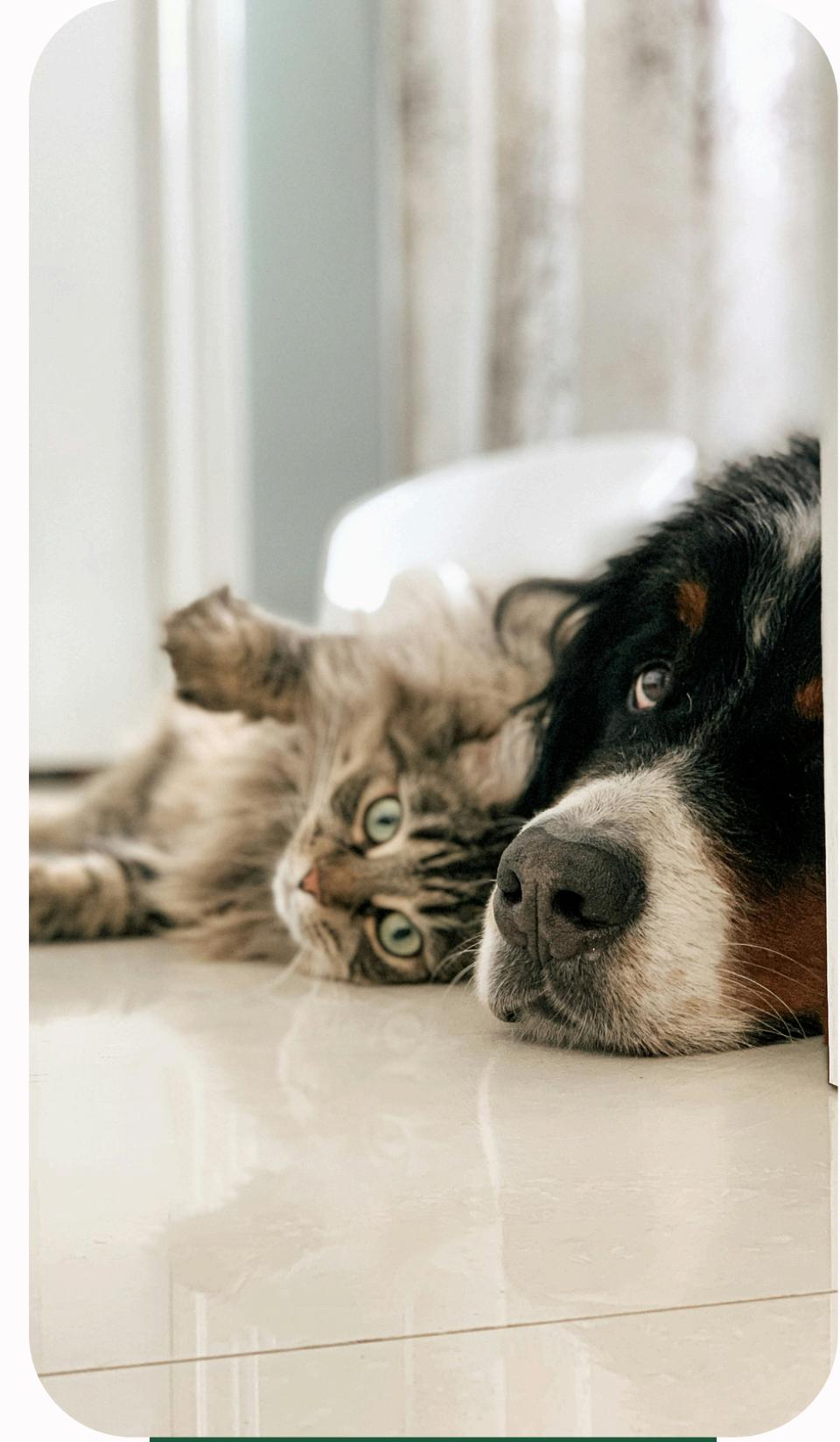


Portfolio Project

Ori&Nori

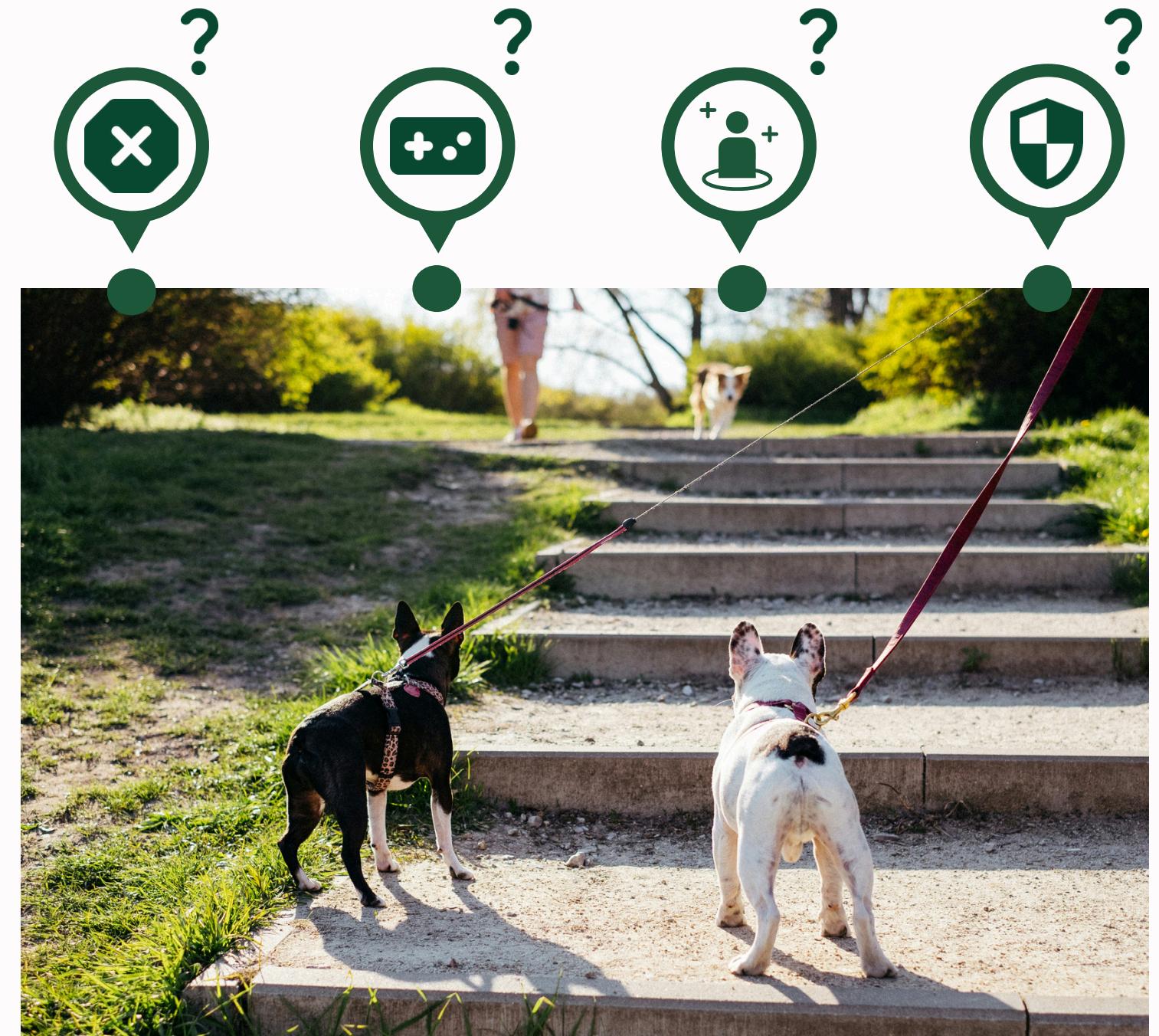
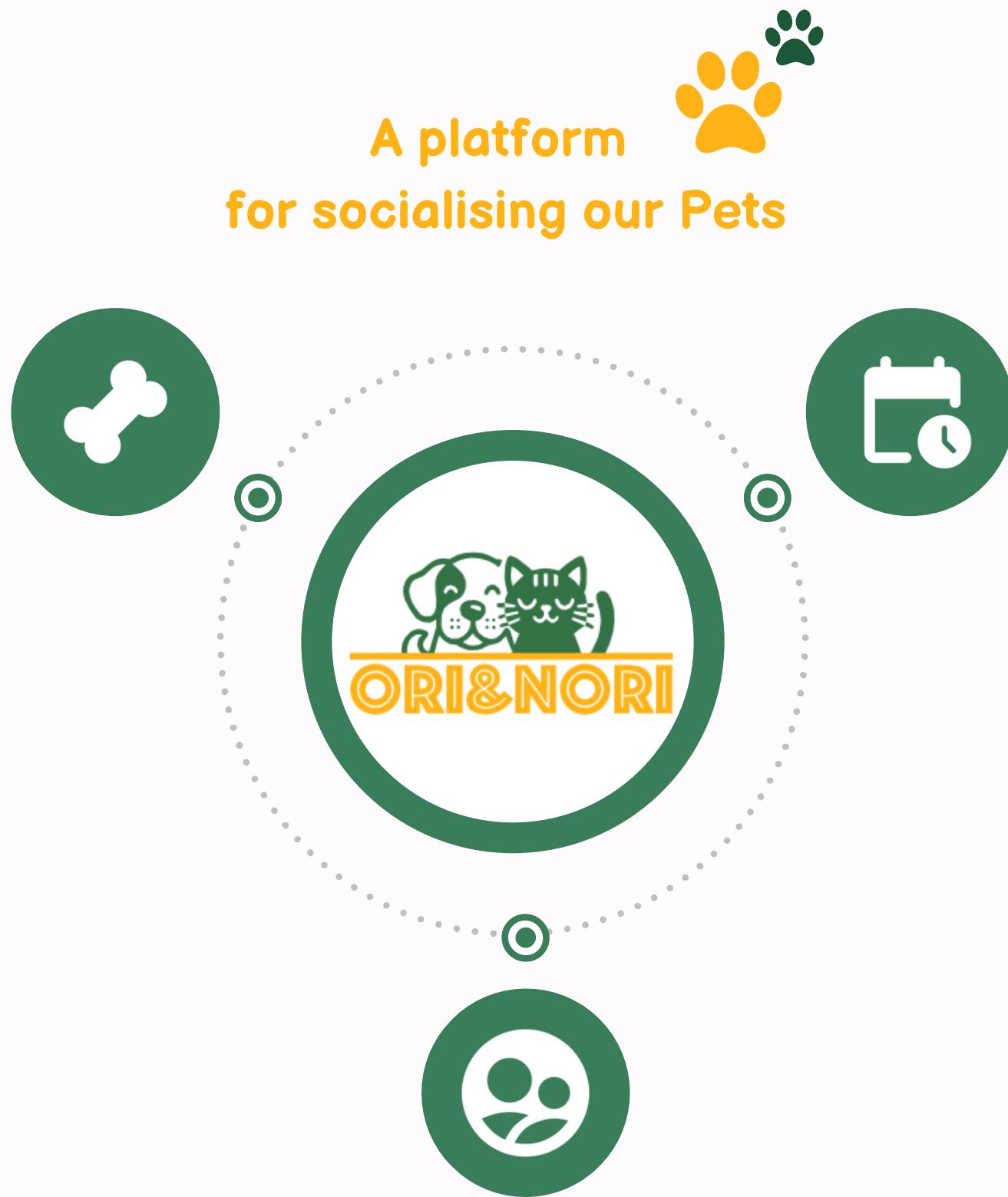


A platform for socialising our Pets



Maël Ezanic & Thomas Mayé #C24

Socialize your Pet : a Challenge





Our purpose

Improve

our companions social lives

Help

their owners easily organize meetups JUL 17

Create

a trusted community of pet owners shake hands

Our Team



Maël Ezanic

BackEnd
& Nori Owner



Thomas Mayé

FrontEnd
Project Manager
& Ori Owner



Features



User

Create, Manage,
Search a Profile



Pet

Create, Manage,
Search a Pet



Filter

Select the species and
breeds you wish to
meet



Meetup

Organize, Manage,
Join lonely or with your
pets



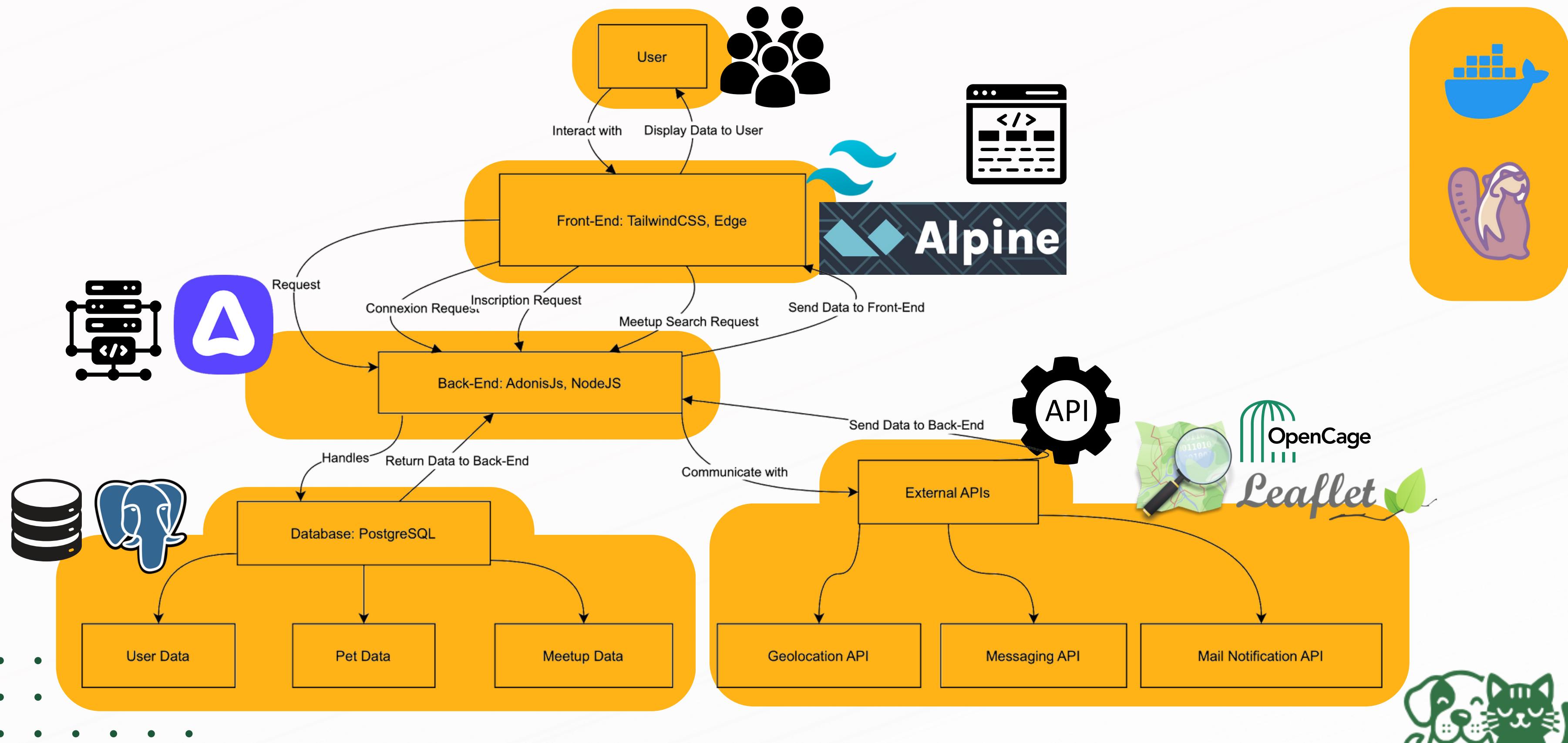
Mapping

Find the meetup
around



DEMO

A Modern Stack



```

/**
 * -----
 * Displays list of available meetups with their locations on a map
 * -----
 */
async displayMeetupsList({ view, auth, response, session }: HttpContext) {
  const user = auth.user
  if (!user) {
    return response.unauthorized({ message: 'User not authenticated' })
  }
  // Get meetups user hasn't joined yet
  const meetups = await Meetup.query()
    .whereDoesntHave('meetupUsers', (query) => {
      query.where('user_id', user.id)
    })
    .preload('meetupPets', (query) => {
      query.preload('breed')
      query.preload('species')
    })
    .orderBy('date', 'asc')

  // Format dates for display
  const formattedMeetups = meetups.map((meetup) => ({
    ...meetup.serialize(),
    formattedDate: DateTime.isDateTime(meetup.date)
      ? meetup.date
      : DateTime.fromJSDate(new Date(meetup.date)).toFormat('dd/MM/yyyy HH:mm'),
  }))

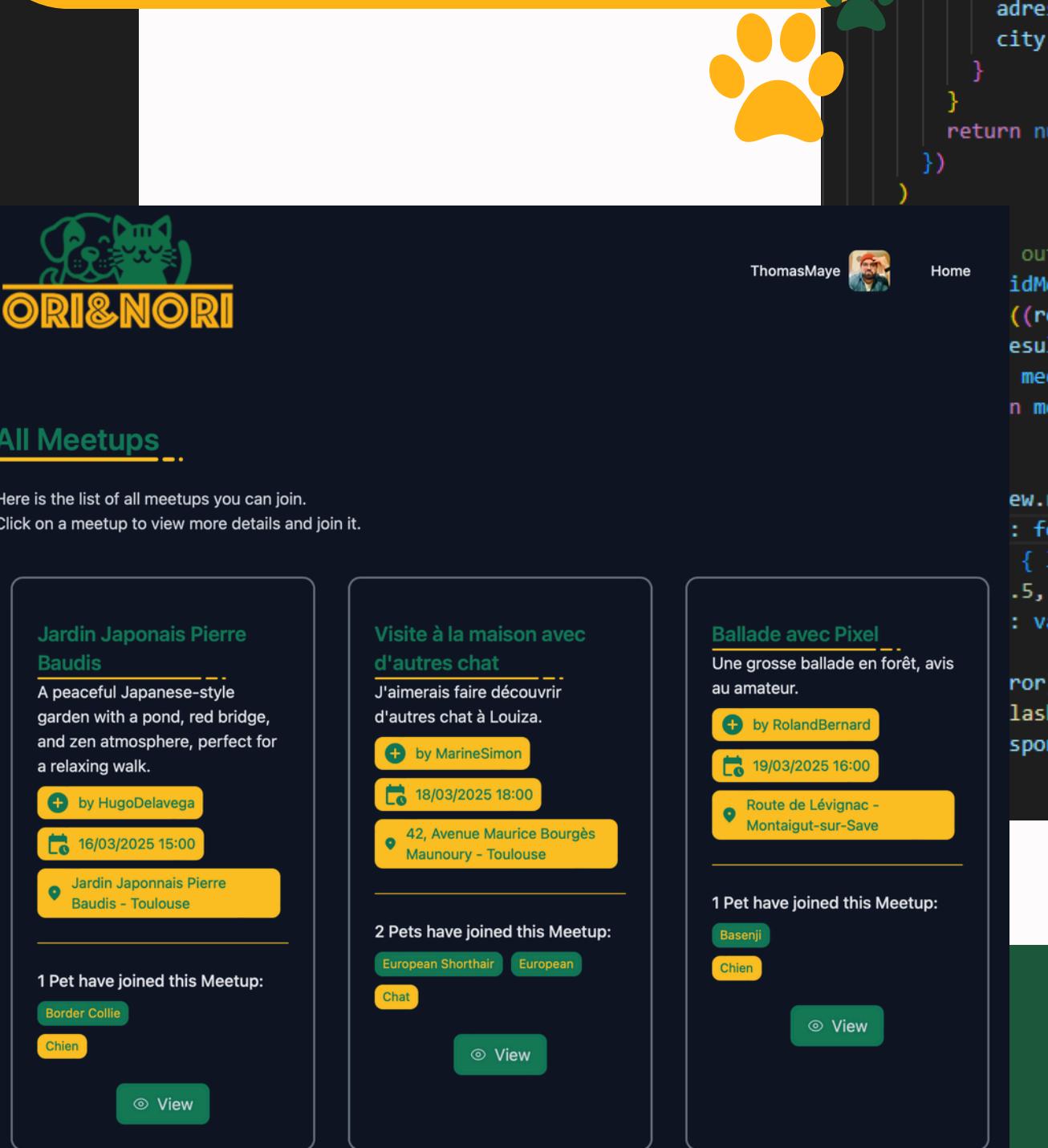
  const apiKey = process.env.OPENCAGE_API_KEY

  if (!apiKey) {
    session.flash('error', 'API key is missing')
    return response.redirect().back()
  }

  // Geocode addresses to get coordinates
  const addresses = meetups.map((meetup) => `${meetup.adress}, ${meetup.city}`)
  try {
    // Process all geocoding requests in parallel
    const coordinates = await Promise.allSettled(
      addresses.map(async (address, index) => {
        const addressData = await opencage.geocode({
          q: address,
          key: apiKey,
        })
      })
    )
  }
}

```

Code snippet



The screenshot shows the Ori&Nori website interface. At the top, there's a navigation bar with a logo featuring a dog and cat, and the text "ORI&NORI". On the right side of the nav bar, there are user profile icons for "ThomasMaye" and a "Home" link. Below the nav bar, the main content area has a dark background with white text. It features a heading "All Meetups" and a sub-instruction: "Here is the list of all meetups you can join. Click on a meetup to view more details and join it." Three meetups are listed in cards:

- Jardin Japonais Pierre Baudis**: A peaceful Japanese-style garden with a pond, red bridge, and zen atmosphere, perfect for a relaxing walk. Organized by HugoDelavega on 16/03/2025 at 15:00. Location: Jardin Japonais Pierre Baudis - Toulouse.
- Visite à la maison avec d'autres chat**: J'aimerais faire découvrir d'autres chat à Louiza. Organized by MarineSimon on 18/03/2025 at 18:00. Location: 42, Avenue Maurice Bourgès Maunoury - Toulouse.
- Ballade avec Pixel**: Une grosse ballade en forêt, avis au amateur. Organized by RolandBernard on 19/03/2025 at 16:00. Location: Route de Lévignac - Montaigut-sur-Save.

Each card also shows the number of pets that have joined the meetup (1 Pet for the first two, 2 Pets for the third), the breeds of the pets (Border Collie, Chien for the first two, Basenji, Chien for the third), and a "View" button.

```

try {
  // Process all geocoding requests in parallel
  const coordinates = await Promise.allSettled(
    addresses.map(async (address, index) => {
      const addressData = await opencage.geocode({
        q: address,
        key: apiKey,
      })
    })
  )
  // Return meetup with coordinates if geocoding successful
  if (addressData && addressData.status.code === 200 && addressData.results.length > 0) {
    const address = addressData.results[0]
    return {
      ...meetups[index],
      latitude: address.geometry.lat,
      longitude: address.geometry.lng,
      title: meetups[index].title,
      date: meetups[index].date,
      description: meetups[index].description,
      address: meetups[index].address,
      city: meetups[index].city,
    }
  }
  return null
}

out failed geocoding attempts
idMeetups = coordinates
((result) => result.status === 'fulfilled' && result.value !== null)
result) => {
  meetup = (result as PromiseFulfilledResult<any>).value
  in meetup
}

ew.render('pages/meetup/meetups', [
  : formattedMeetups,
  { latitude: 46.729, longitude: 2.505 }, // Saint-Amand-Montrond
  .5,
  : validMeetups,
  ror) {
  lash('error', 'Error fetching coordinates')
  response.redirect().back()
}

```

```

/**
 * -----
 * Displays list of available meetups with their locations on a map
 * -----
 */
async displayMeetupsList({ view, auth, response, session }: HttpContext) {
  const user = auth.user
  if (!user) {
    return response.unauthorized({ message: 'User not authenticated' })
  }
  // Get meetups user hasn't joined yet
  const meetups = await Meetup.query()
    .whereDoesntHave('meetupUsers', (query) => {
      query.where('user_id', user.id)
    })
    .preload('meetupPets', (query) => {
      query.preload('breed')
      query.preload('species')
    })
    .orderBy('date', 'asc')

  // Format dates for display
  const formattedMeetups = meetups.map((meetup) => ({
    ...meetup.serialize(),
    formattedDate: DateTime.isDateTime(meetup.date)
      ? meetup.date
      : DateTime.fromJSDate(new Date(meetup.date)).toFormat('dd/MM/yyyy HH:mm'),
  }))

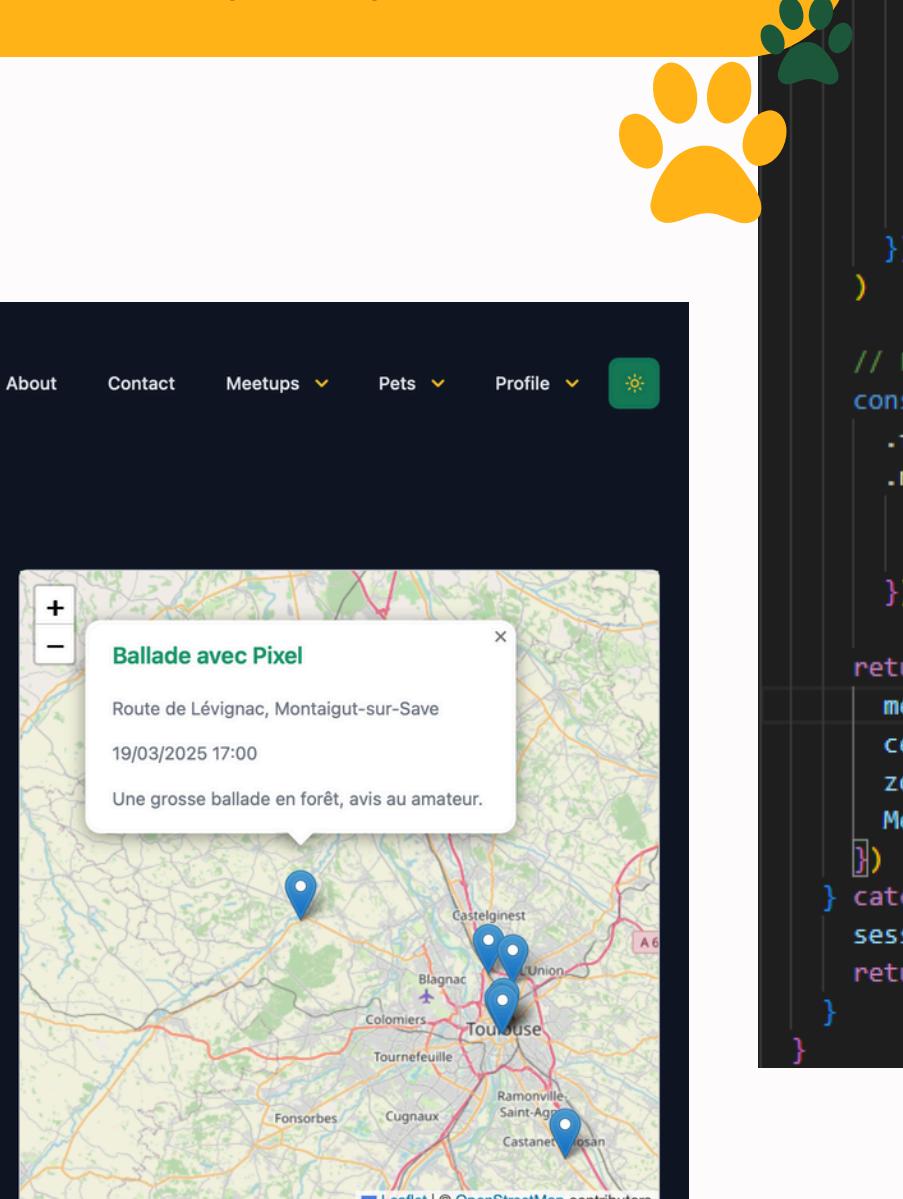
  const apiKey = process.env.OPENCAGE_API_KEY

  if (!apiKey) {
    session.flash('error', 'API key is missing')
    return response.redirect().back()
  }

  // Geocode addresses to get coordinates
  const addresses = meetups.map((meetup) => `${meetup.adress}, ${meetup.city}`)
  try {
    // Process all geocoding requests in parallel
    const coordinates = await Promise.allSettled(
      addresses.map(async (address, index) => {
        const addressData = await opencage.geocode({
          q: address,
          key: apiKey,
        })
      })
    )
  }
}

```

Code snippet



```

try {
  // Process all geocoding requests in parallel
  const coordinates = await Promise.allSettled(
    addresses.map(async (address, index) => {
      const addressData = await opencage.geocode({
        q: address,
        key: apiKey,
      })
    })
  )
  // Return meetup with coordinates if geocoding successful
  if (addressData && addressData.status.code === 200 && addressData.results.length > 0) {
    const address = addressData.results[0]
    return {
      ...meetups[index],
      latitude: address.geometry.lat,
      longitude: address.geometry.lng,
      title: meetups[index].title,
      date: meetups[index].date,
      description: meetups[index].description,
      address: meetups[index].address,
      city: meetups[index].city,
    }
  }
  return null
}

// Filter out failed geocoding attempts
const validMeetups = coordinates
  .filter((result) => result.status === 'fulfilled' && result.value !== null)
  .map((result) => {
    const meetup = (result as PromiseFulfilledResult<any>).value
    return meetup
  })

return view.render('pages/meetup/meetups', {
  meetups: formattedMeetups,
  center: { latitude: 46.729, longitude: 2.505 }, // Saint-Amand-Montrond
  zoom: 5.5,
  Meetups: validMeetups,
})
} catch (error) {
  session.flash('error', 'Error fetching coordinates')
  return response.redirect().back()
}
}

```

```

/** 
* -----
* Displays list of available meetups with their locations on a map
* -----
*/
async displayMeetupsList({ view, auth, response, session }: HttpContext) {
  const user = auth.user
  if (!user) {
    return response.unauthorized({ message: 'User not authenticated' })
  }
  // Get meetups user hasn't joined yet
  const meetups = await Meetup.query()
    .whereDoesntHave('meetupUsers', (query) => {
      query.where('user_id', user.id)
    })
    .preload('meetupPets', (query) => {
      query.preload('breed')
      query.preload('species')
    })
    .orderBy('date', 'asc')

  // Format dates for display
  const formattedMeetups = meetups.map((meetup) => ({
    ...meetup.serialize(),
    formattedDate: DateTime.isDateTime(meetup.date)
      ? meetup.date
      : DateTime.fromJSDate(new Date(meetup.date)).toFormat(
        'dd/MM/yyyy HH:mm'
      )
  }))

  const apiKey = process.env.OPENCAGE_API_KEY

  if (!apiKey) {
    session.flash('error', 'API key is missing')
    return response.redirect().back()
  }

  // Geocode addresses to get coordinates
  const addresses = meetups.map((meetup) => `${meetup.adress}`)
  try {
    // Process all geocoding requests in parallel
    const coordinates = await Promise.allSettled(
      addresses.map(async (address, index) => {
        const addressData = await opencage.geocode({
          q: address,
          key: apiKey,
        })
      })
    )
  }
}

```

Code snippet



All Meetups

Here is the list of all meetups you can join.
Click on a meetup to view more details and join it.

Jardin Japonais Pierre Baudis
A peaceful Japanese-style garden with a pond, red bridge, and zen atmosphere, perfect for a relaxing walk.

by HugoDelavega
16/03/2025 15:00
Jardin Japonais Pierre Baudis - Toulouse

1 Pet have joined this Meetup:
Border Collie
Chien

[View](#)

Visite à la maison avec d'autres chat
J'aimerais faire découvrir d'autres chat à Louiza.

by MarineSimon
18/03/2025 18:00
42, Avenue Maurice Bourgès Maunoury - Toulouse

2 Pets have joined this Meetup:
European Shorthair
European Chat

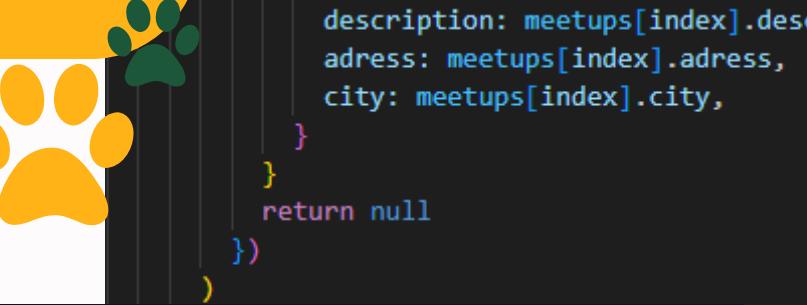
[View](#)

Ballade avec Pixel
Une grosse ballade en forêt, avis au amateur.

by RolandBernard
19/03/2025 16:00
Route de Lévignac - Montaigut-sur-Save

1 Pet have joined this Meetup:
Basenji
Chien

[View](#)



```

try {
  // Process all geocoding requests in parallel
  const coordinates = await Promise.allSettled(
    addresses.map(async (address, index) => {
      const addressData = await opencage.geocode({
        q: address,
        key: apiKey,
      })
    })
  )
  // Return meetup with coordinates if geocoding successful
  if (addressData && addressData.status.code === 200 && addressData.results.length > 0) {
    const address = addressData.results[0]
    return {
      ...meetups[index],
      latitude: address.geometry.lat,
      longitude: address.geometry.lng,
      title: meetups[index].title,
      date: meetups[index].date,
      description: meetups[index].description,
      address: meetups[index].address,
      city: meetups[index].city,
    }
  }
  return null
}

)

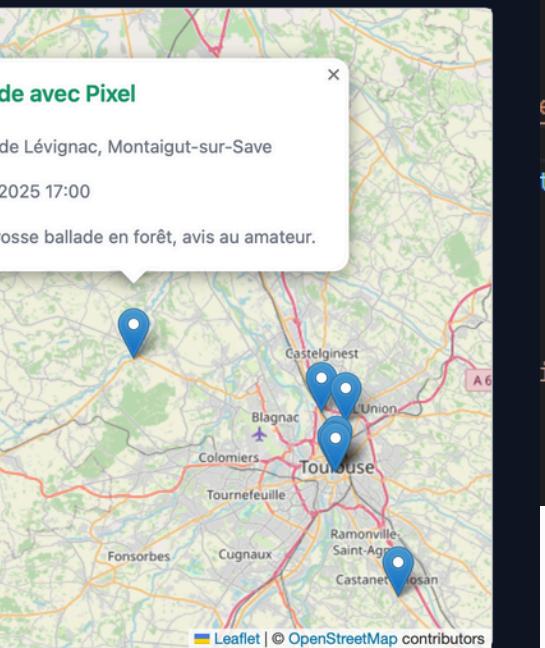
```

```

mpts
    === 'fulfilled' && result.value !== null)

seFulfilledResult<any>.value

```

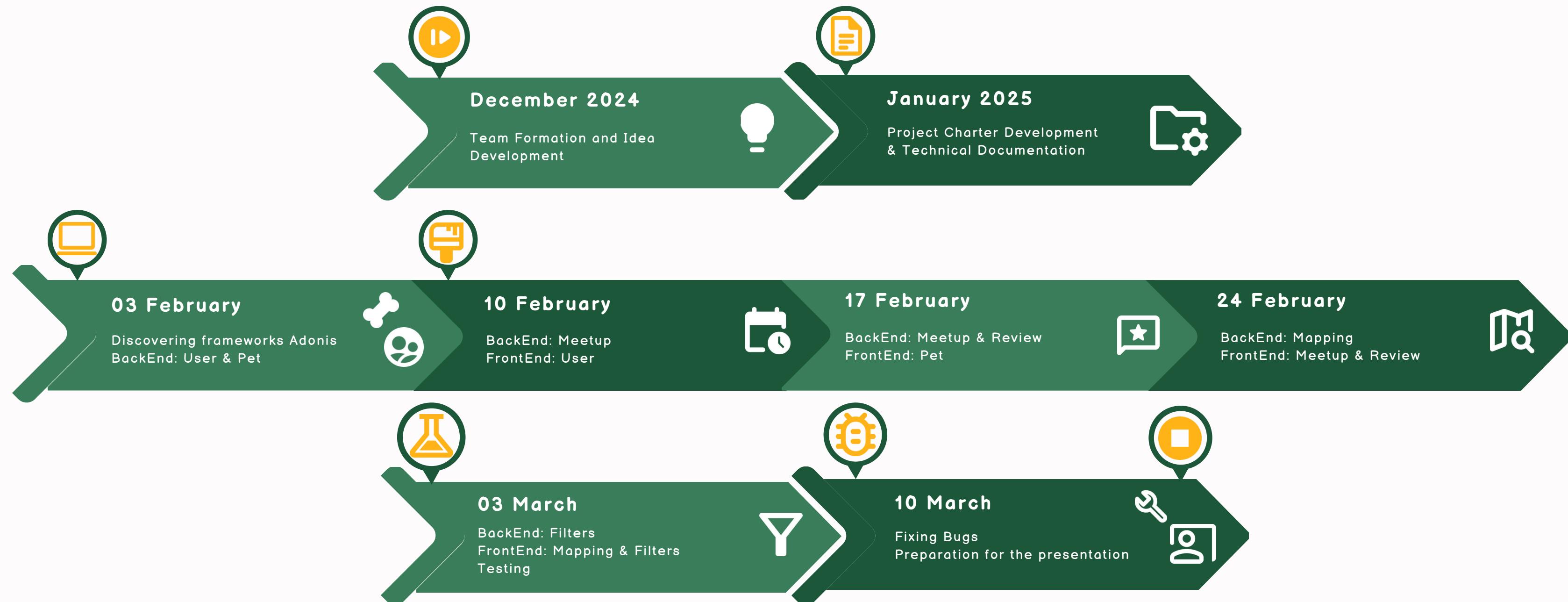


```

etups', [
tude: 2.505 }, // Saint-Amand-Montrond
ing coordinates')

```

Timeline



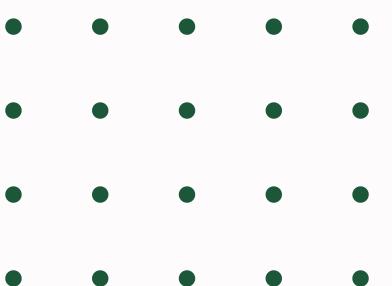
Result

- 01 Successful project
- 02 Technical implementation
- 03 Cohesive design system
- 04 Some technicals challenges
- 05 Unit test
- 06 User experience
- 07 Project Management
- 08 Learning from mistakes





Future Of





THANK YOU

if you have any questions?

