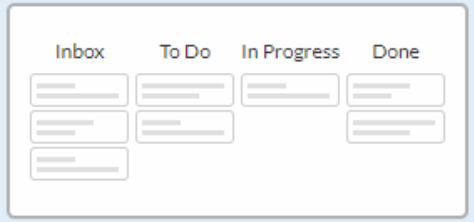
using Waffle to plan features

- Go to Waffle.io https://waffle.io/ .
- Click on Login or Get Started for Free.
- Select Public & Private Repos to allow access to all of your repositories.
- Click on Create Project.
- Search for the local-weather-app repository and select it.
- Hit Continue.

Waffle boards

Choose Board Layout

Your columns are based off your GitHub labels.



Basic

Quick way to get started, you can always customize later.

Advanced

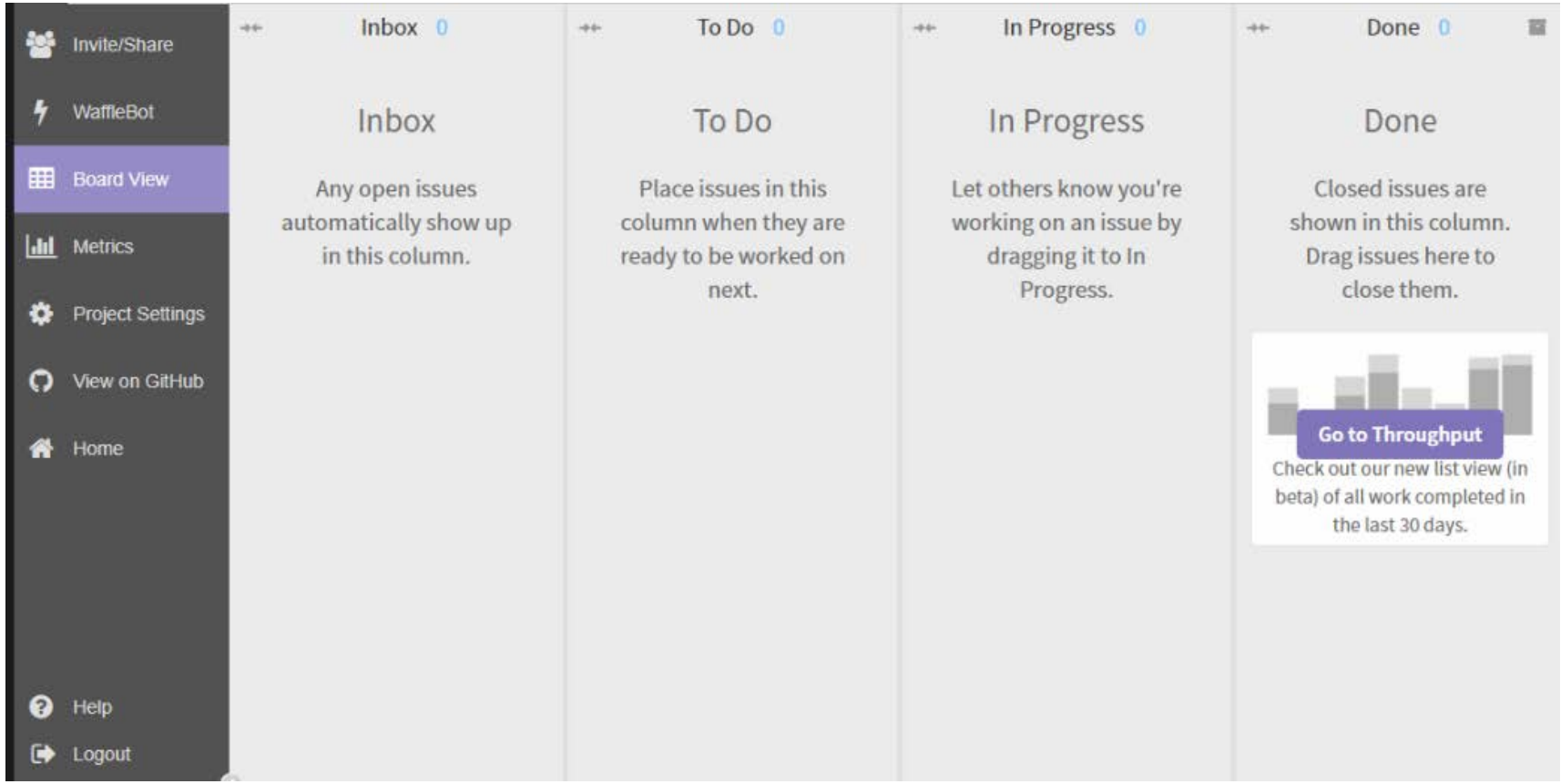
Perfect for teams, additional columns to help your team plan and review.

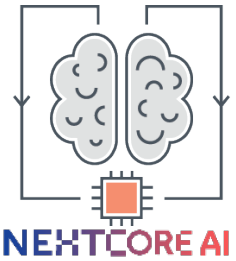
Don't worry, you can customize your layout anytime from your project settings.

Create Project

using Waffle to plan features

- Select the Basic layout and click on Create Project.
- You will see a new board created for you.





Creating issues for Local Weather app

- Display Current Location weather information for the current day
- Display forecast information for current location
- Add city search capability so that users can see weather information from other cities
- Add a preferences pane to store the default city for the user
- Improve the UX of the app with Angular Material

UI elements

src/app/app.component.html

```
<div style="text-align:center">
  <h1>
  LocalCast Weather
  </h1>
  <div>Your city, your forecast, right now!</div>
  <h2>Current Weather</h2>
  <div>current weather</div>
</div>
```

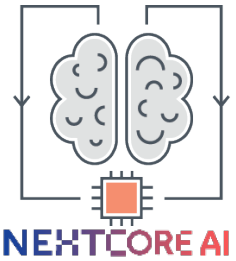
At this point, you should run npm start and navigate to http://localhost:5000 on your browser so that you can observe the changes you're making in real time.

Adding an Angular component

- need to display the current weather information, where `<div>current weather</div>` is locatedDisplay forecast information for current location

- In the terminal, execute `npx ng generate component current-weather`
- Observe the new files created in your app folder:

```
src/app
├── app.component.css
├── app.component.html
├── app.component.spec.ts
├── app.component.ts
├── app.module.ts
├── current-weather
  ├── current-weather.component.css
  ├── current-weather.component.html
  ├── current-weather.component.spec.ts
  └── current-weather.component.ts
```



Creating issues for Local Weather app

- A generated component has four parts:

- current-weather.component.css contains any CSS that is specific to the component and is an optional file
- current-weather.component.html contains the HTML template that defines the look of the component and rendering of the bindings, and can be considered the View, in combination with any CSS styles used
- current-weather.component.spec.ts contains Jasmine-based unit tests that you can extend to test your component functionality
- current-weather.component.ts contains the @Component decorator above the class definition and is the glue that ties together the CSS, HTML, and JavaScript code together. The class itself can be considered the ViewModel, pulling data from services and performing any necessary transformations to expose sensible bindings for the View, as shown as follows: