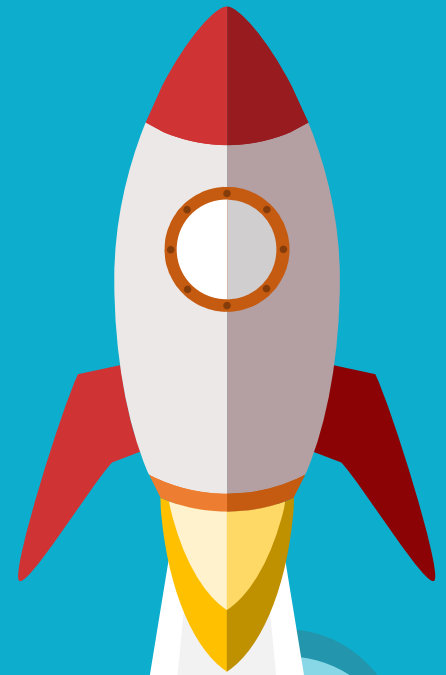




# Final Project

Kick Off !



# Project Planning



# Project Description

## What's the purpose of your app?

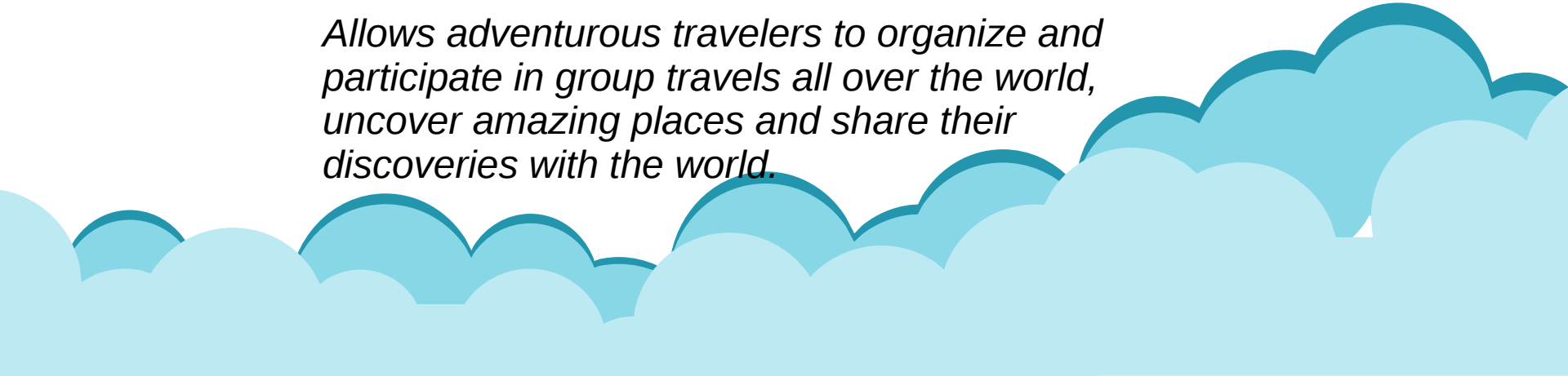
- What problem does your app solves?
- Who is your target audience?
  - Type of user
  - Geography
  - Age

You should be able to describe your app in a sentence or a short paragraph.

Ex:

*Create the travel of your life!*

*Allows adventurous travelers to organize and participate in group travels all over the world, uncover amazing places and share their discoveries with the world.*



# Defining an MVP (Minimum Viable Product)

A minimum viable product has just those *core features sufficient to deploy* the product, and no more

- Get the product to early customers as soon as possible
- Test the product before investing more
- Base future development on feedback from users

For the bootcamp:

- Develop the core features to showcase the app to potential employers.



# Defining an MVP (Minimum Viable Product)

- List of core features
- List of additional features
- Requirements (ex. Does it need to be responsive?)



# User Stories

1. Assuming you know what's best and ignoring the user



2. Doing exactly what the user ask

Design your app by focusing on users' needs!

# User Stories

**User should be able to add an event to the group trip calendar**

As a user I want to add an event to the group calendar because I want to invite my fellow travelers to do something cool together such visiting a sight worth seeing or participate in an activity.



# User Stories

## Title:

User should be able to add an event to the group trip calendar

## User Story:

As a user I want to add a event to the group calendar because I want to invite my fellow travelers to do something interesting together such visiting a sight worth seeing or participate in an activity.

- User **should**... in the title
- As a user, I want to (goals, desire) because (benefit)

Additional reading:

A framework for modern User Stories

<https://medium.com/@jonatisokon/a-framework-for-user-stories-bc3dc323eca9>





# User Stories

## Who is the user?

- Create a user persona



### Demographics

- 27 years old
- Single
- Earns 85k a year

### Goals

- Build a strong portfolio
- Get involved in a large project
- Acquire more skills

### Behaviors & Beliefs

- Obsessive about code refactoring
- Very details oriented
- Wants to be involved in the development process

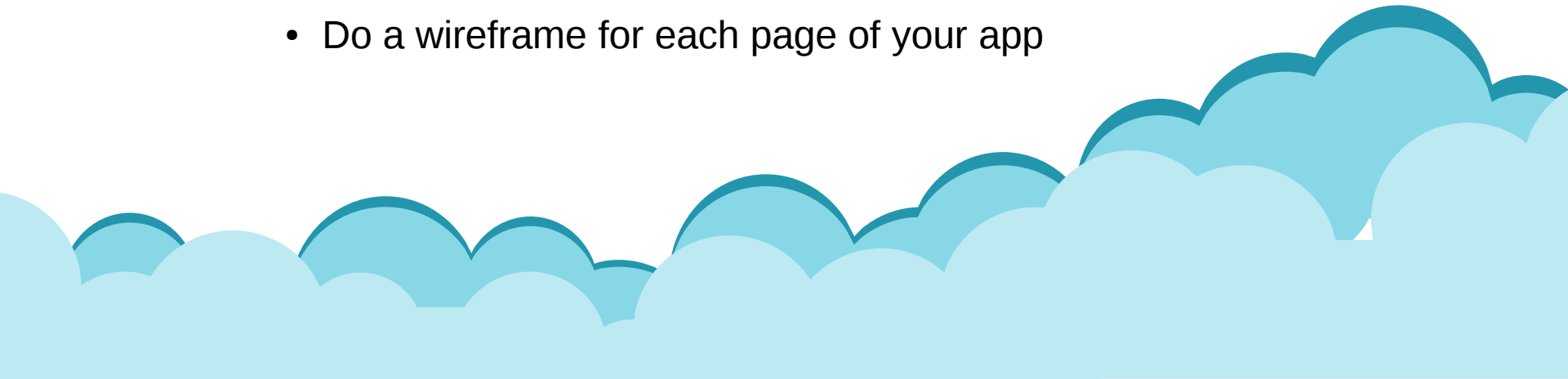
### Attributes

- Tech Savviness: 5
- Education: 4
- Ambition: 5
- Experience: 3

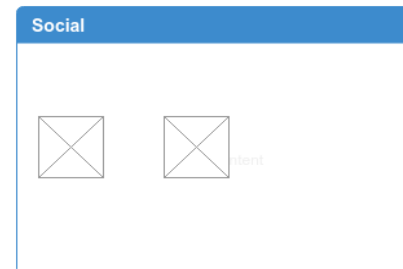
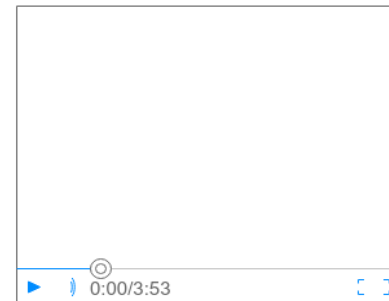
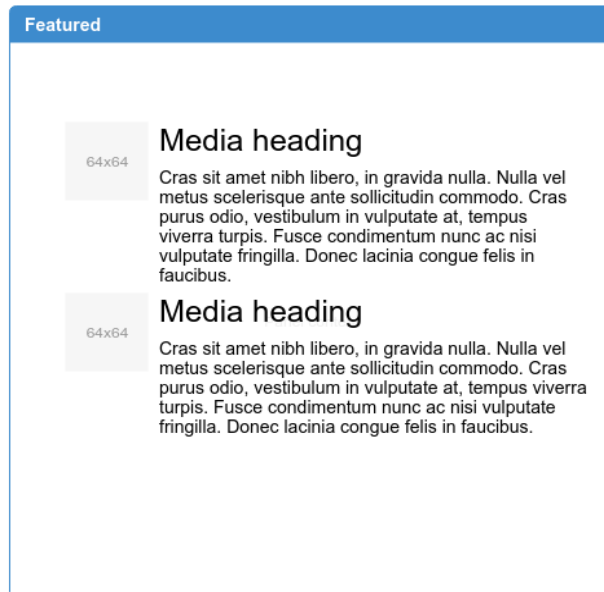
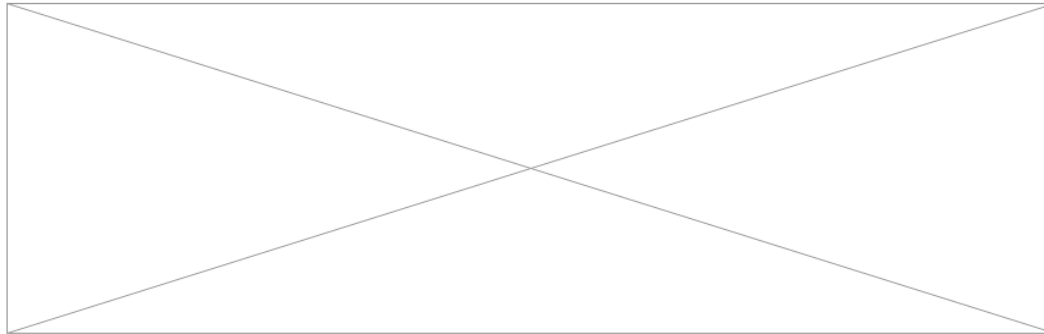
# Wireframes

**A Wireframe is a simplified visual representation of the user interface**

- Based on the user flow from the user stories
- Paper, white board => more flexibility
- Design with an app: easier to get back to, share, and modify
- Use what is simple. Don't spend time on complicated software. (draw.io, mockflow, balsamiq)
- Do a wireframe for each page of your app



# Wireframes



# Technology Stack

## Front End



## Back End



## Choosing Your Front-End

### UI Framework

- Bootstrap, Foundation, Material UI, Sementic UI
- Responsiveness?
- Sass, JQuery?

### Front-End Framework

- React, Angular, VueJS, Meteor

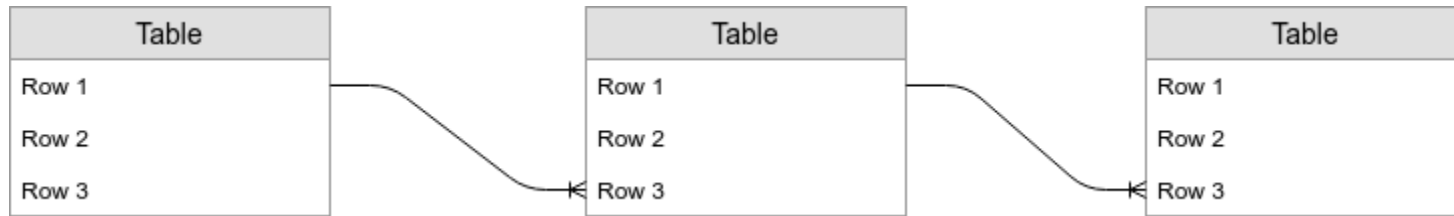
## Choosing Your Back-End

- Rails, Express, Sails

## Choosing Your Database

- SQLite, Postgresql, Mongodb

# ERD



- What data does your app need?
- Use a software (draw.io, lucidChart, etc)

# Project Workflow

Project Planning Document in Compass



# Project Milestones



## Create list of project milestones

- Be specific
- Have deadlines

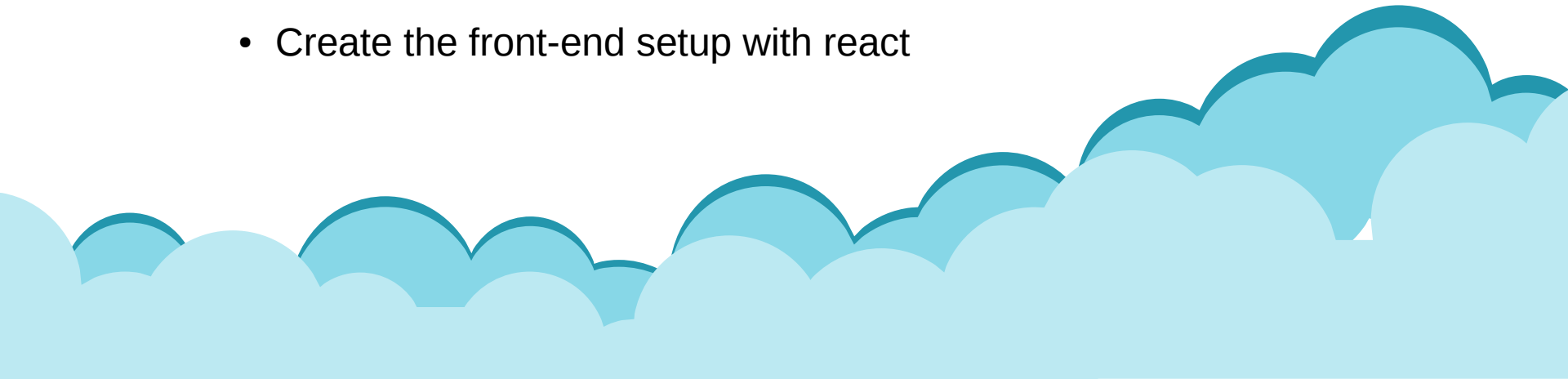


## Example:

Dayboard

### **Have the project setup by Thursday 11 am:**

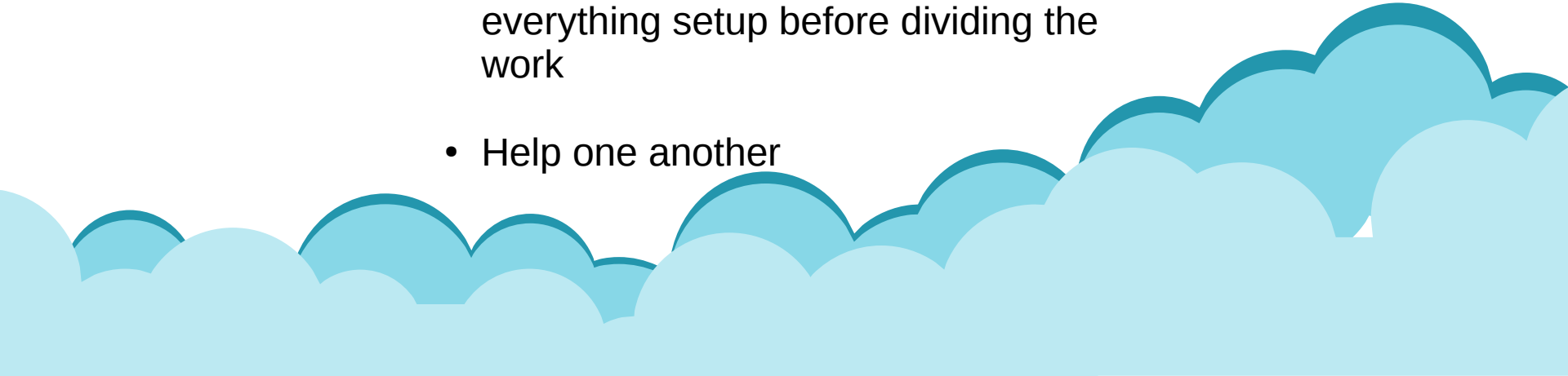
- Create the initial migrations and seed file for the database based on the ERD
- Create the back-end setup with express and knex
- Create the front-end setup with react



# Distribute Work

- Distribute work by feature?
- Preference for front-end, back-end?
- Some pair coding?

## Keep in mind:

- Work should be equally divided
  - You might want to consider having everything setup before dividing the work
  - Help one another
- 

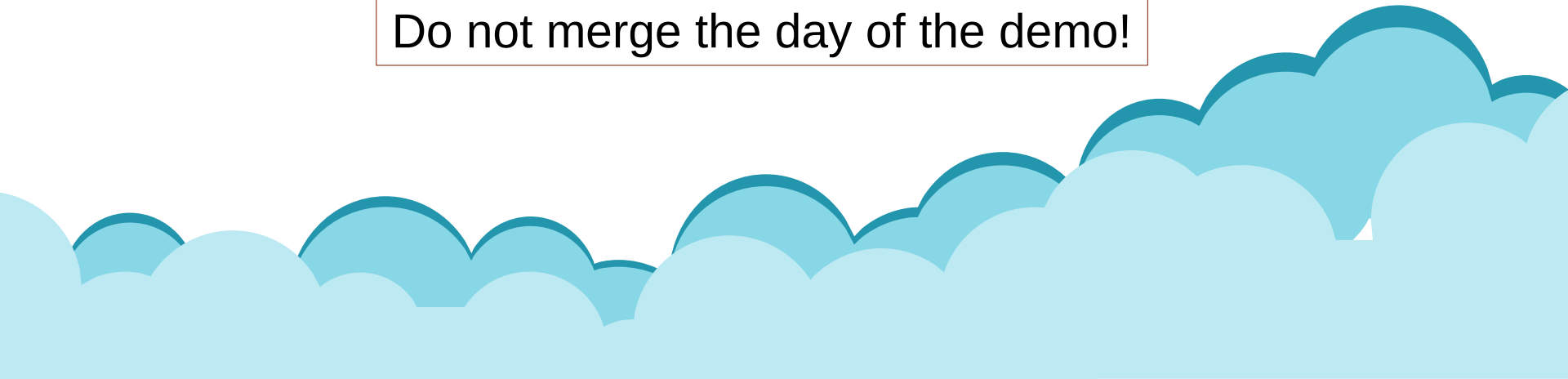


# How to Work

Agree on:

- Coding Style to ensure consistency (prettier, eslint)
- Please use keyboard shortcuts!
- GIT Workflow
  - Never code on master!
  - Feature workflow
  - How to merge

Do not merge the day of the demo!



# Communication

Communication skills are as important as technical skills



# Daily Stand-Ups

In front of the *stand up master*, each team member will disclose:

- What has been accomplished
- What they will be working on
- What hurdles they are facing

Improve communication and  
better follow-ups



# Development



# Project Setup

## Scaffolding

- Use an existing boilerplate
- Use a generator
- Front-End & Back-End Scaffolding
  - Build static page at first?
  - Make CSS work
  - Think about migrations, seeds, validations



# Routing

- Front-end and Back-end routing
- Design your routes according to REST
- API on the back-end?
  - /api/v1/
  - Test your API with Curl or Postman



# Deployment

- Have a working version ready on localhost
- Decide on what computer you will demo the app
- Have a backup computer ready

## Deploy Heroku

- Make sure you have the time
- Work on a separate branch



# Things to keep in mind

- If you try something new, focus on just one new thing. Think about documentation and support.
- Diminish uncertainties: test your setup, the DB, migrations, seeds, AJAX calls, test external APIs, etc.
- Async can be tricky!
- Use helper functions
- Don't wait too long before integrating your features

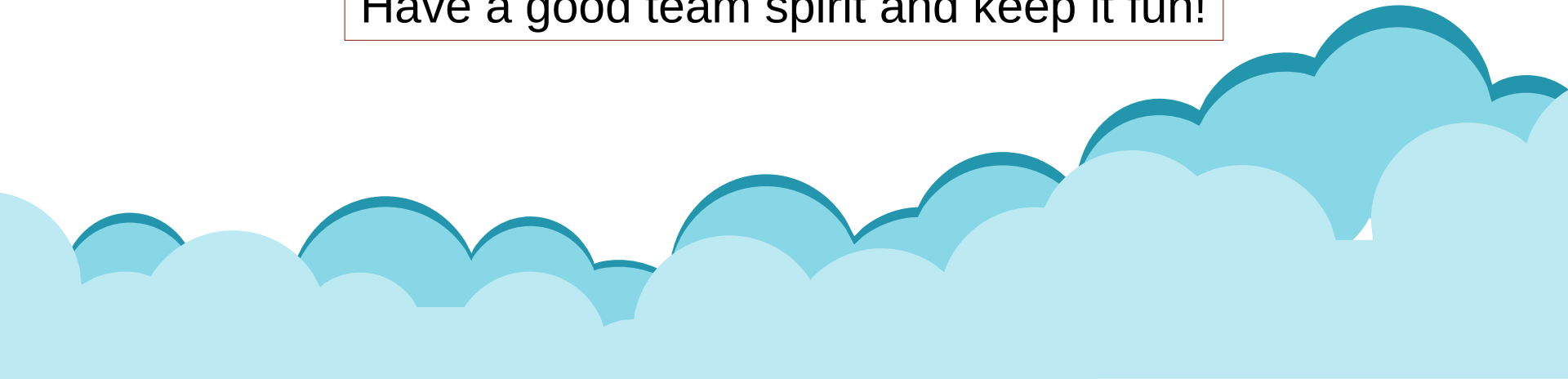




# Things to keep in mind

- Login for SPAs might need to be implemented differently: JWT, oauth
- Focus on the important features. Don't spend a lot of time on features that are not super important (ex. Login)
- Think about the user experience. Have a good UI.

Have a good team spirit and keep it fun!



Questions ?



# Let's do this!

