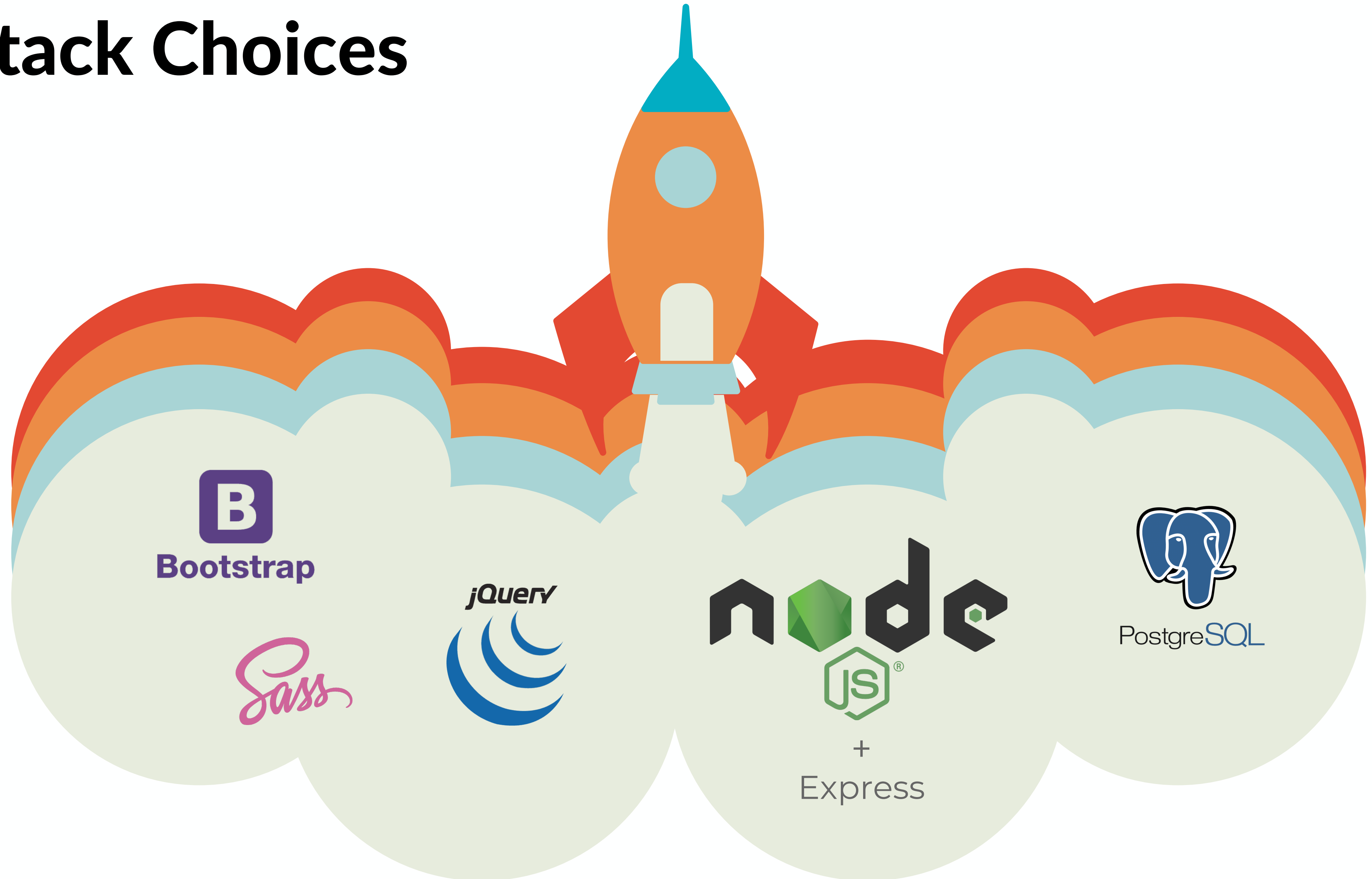


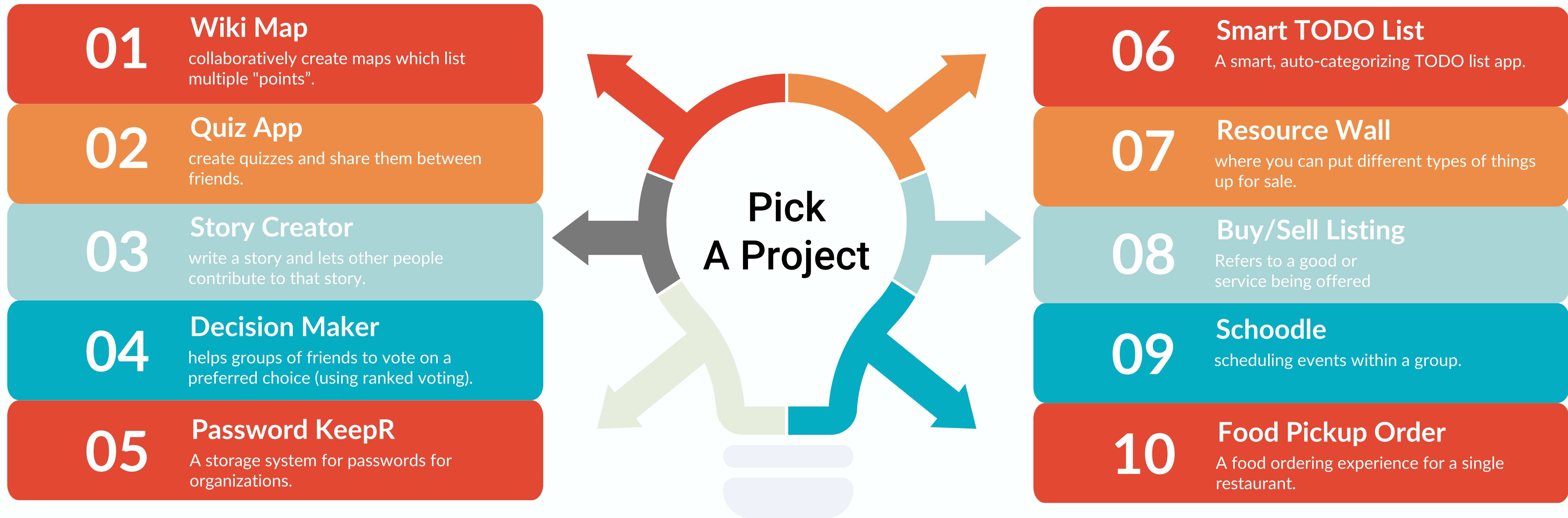
Midterm Project **Kickoff**

Module 6: Week 14 & Week 15

Stack Choices

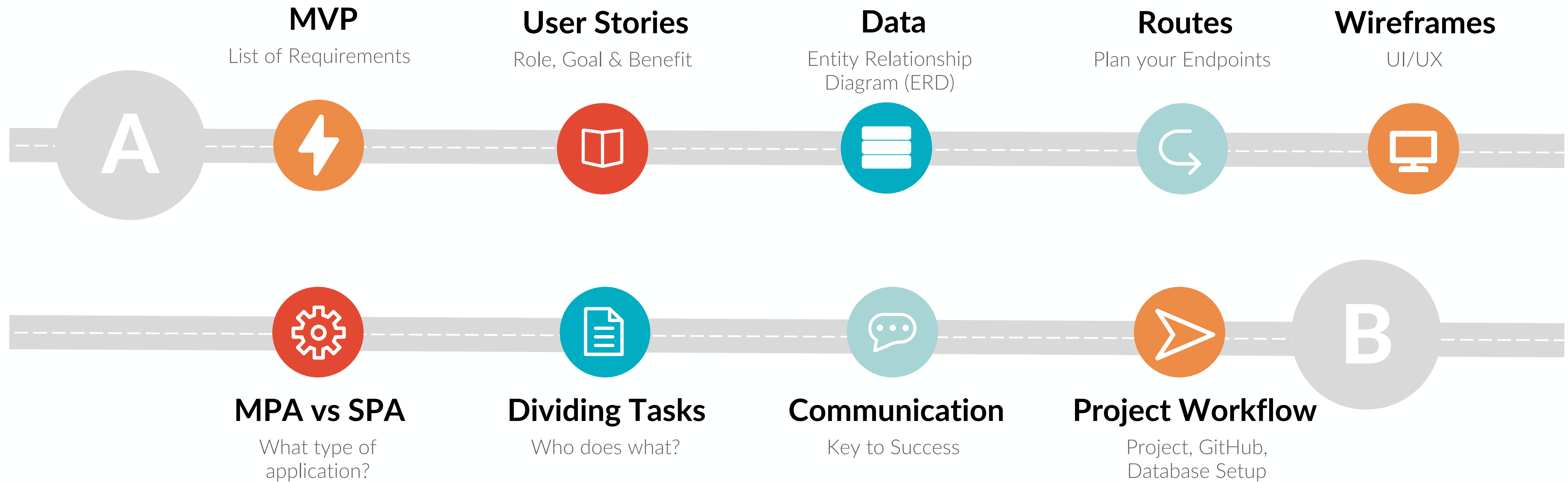


What Do You Want To Learn?



Roadmap

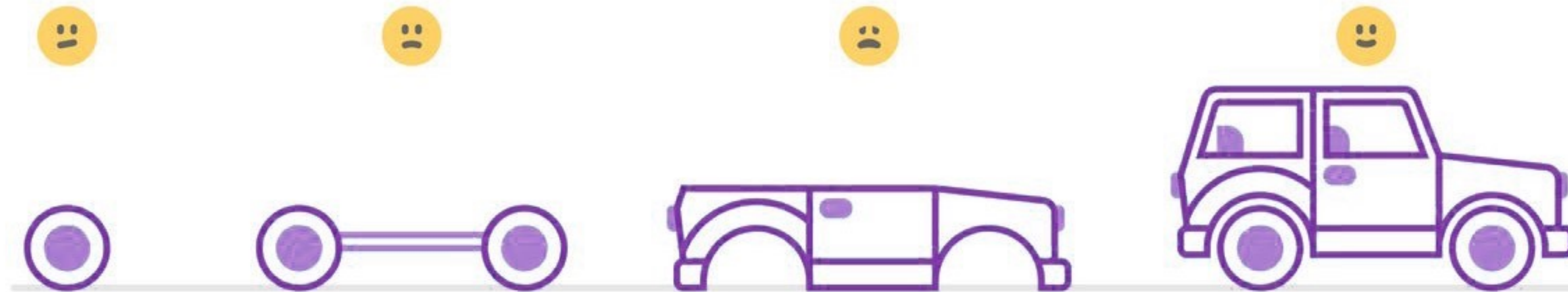
Planning saves time & keeps you organized.



MVP -> MVD

What is the Minimum Viable Product? MVD?

This is NOT MVP...



This IS MVP...



User Stories

Role, Goal & Benefit

What is a User Story?

“A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer. The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer.” - Atlassian

(Role) As a User...

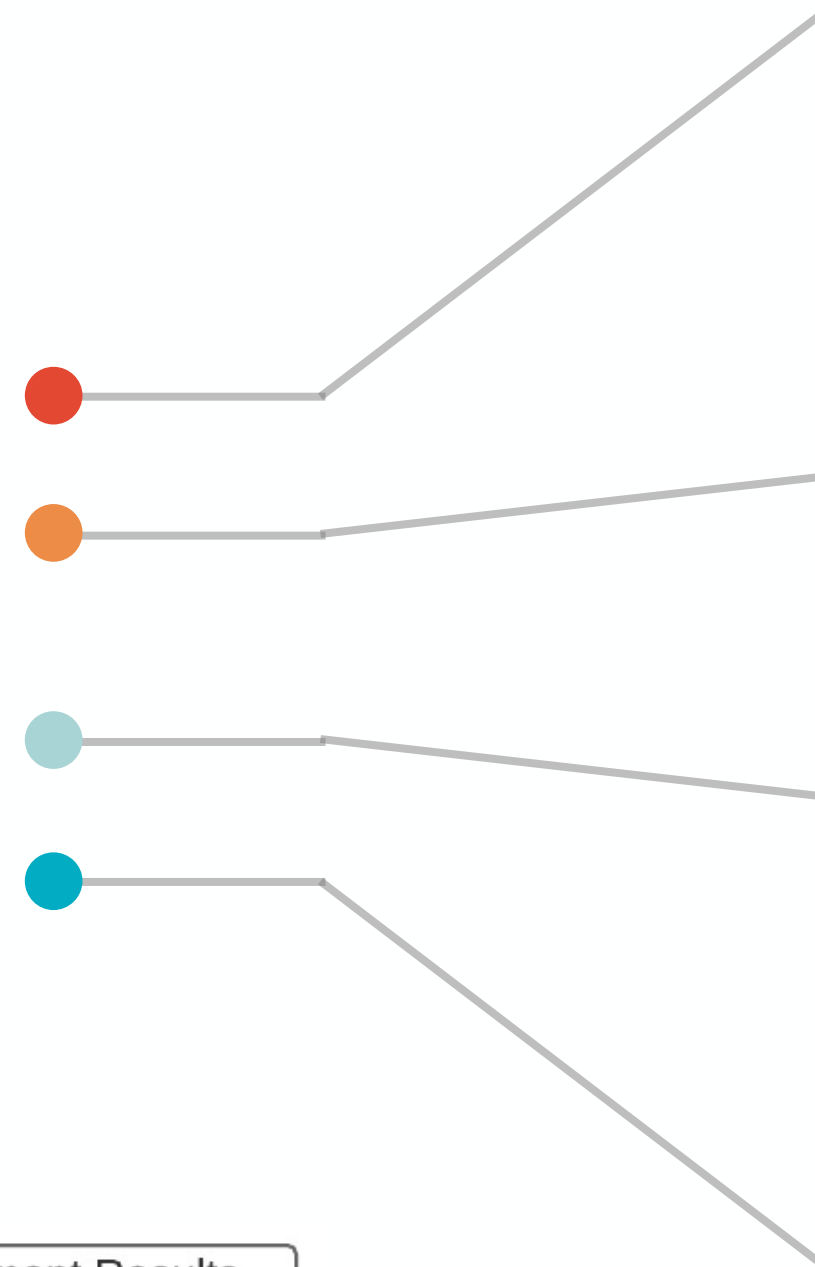
(Goal) I want to...

(Benefit) Because...

As a logged in user, I want to see
all the available maps because I
want to quickly select a map that
matches my interests.

Database

It all starts with data



ERD

Entity Relationship Diagram.

Tables

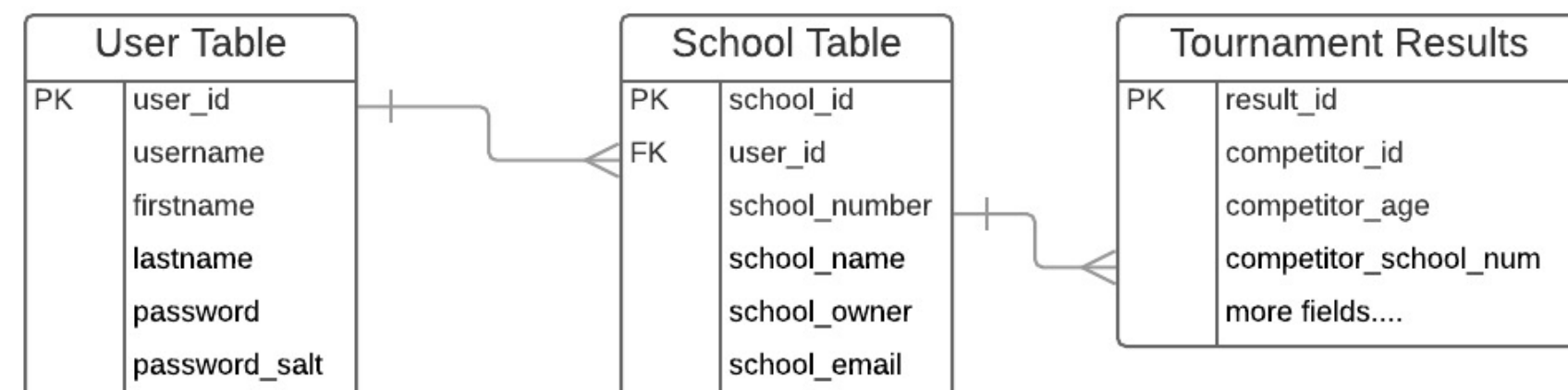
Tables are nouns, they can be taken right from **User Stories**.

Relationships

How are tables connected? What is the data you are accessing or getting?

Tools

Draw.io, Figma, pen & paper...etc.



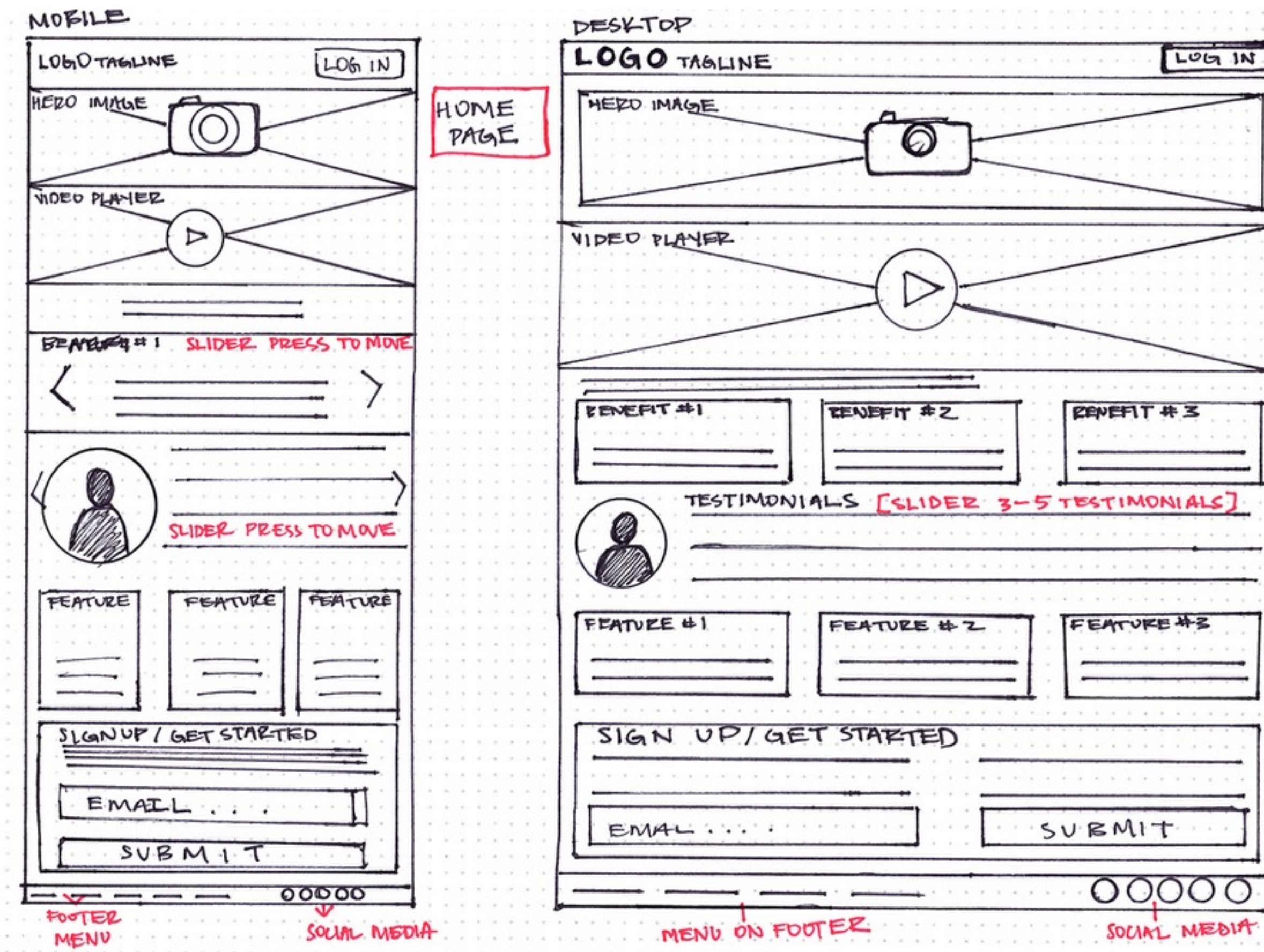
Routes

Planning your Endpoints

HTTP method	URL pattern	Use
GET	/posts	Show all posts
GET	/posts/new	Display a form for new post
POST	/posts	Create new post
GET	/posts/:id	Show a single post
GET	/posts/:id/edit	Display a form for editing a single post
PATCH or PUT	/posts/:id	Update a single post
DELETE	/posts/:id	Delete a single post

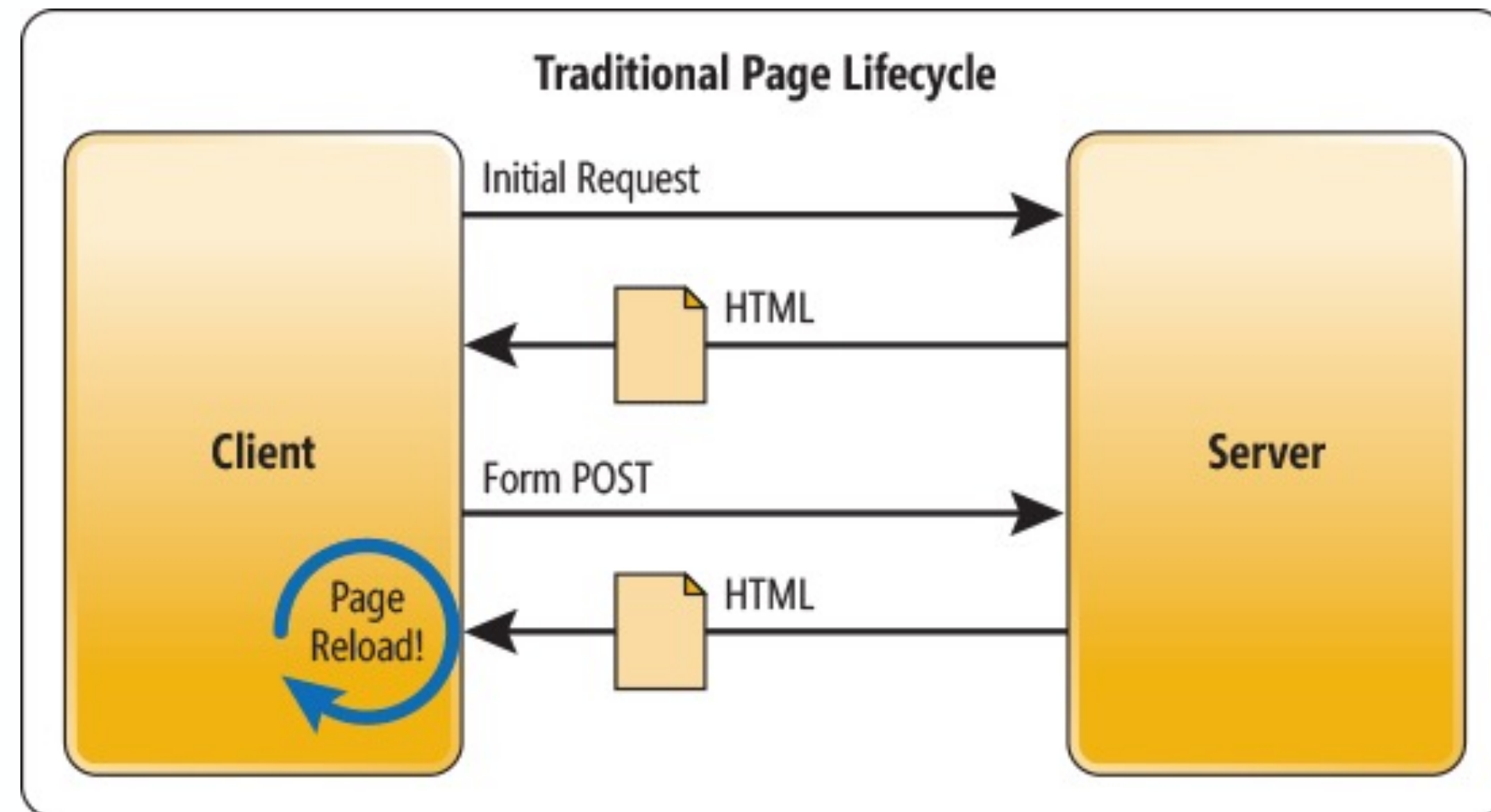
Wireframes

Low-Res Prototype



MPA vs SPA

Multi-Page | Single-Page Application | Combo

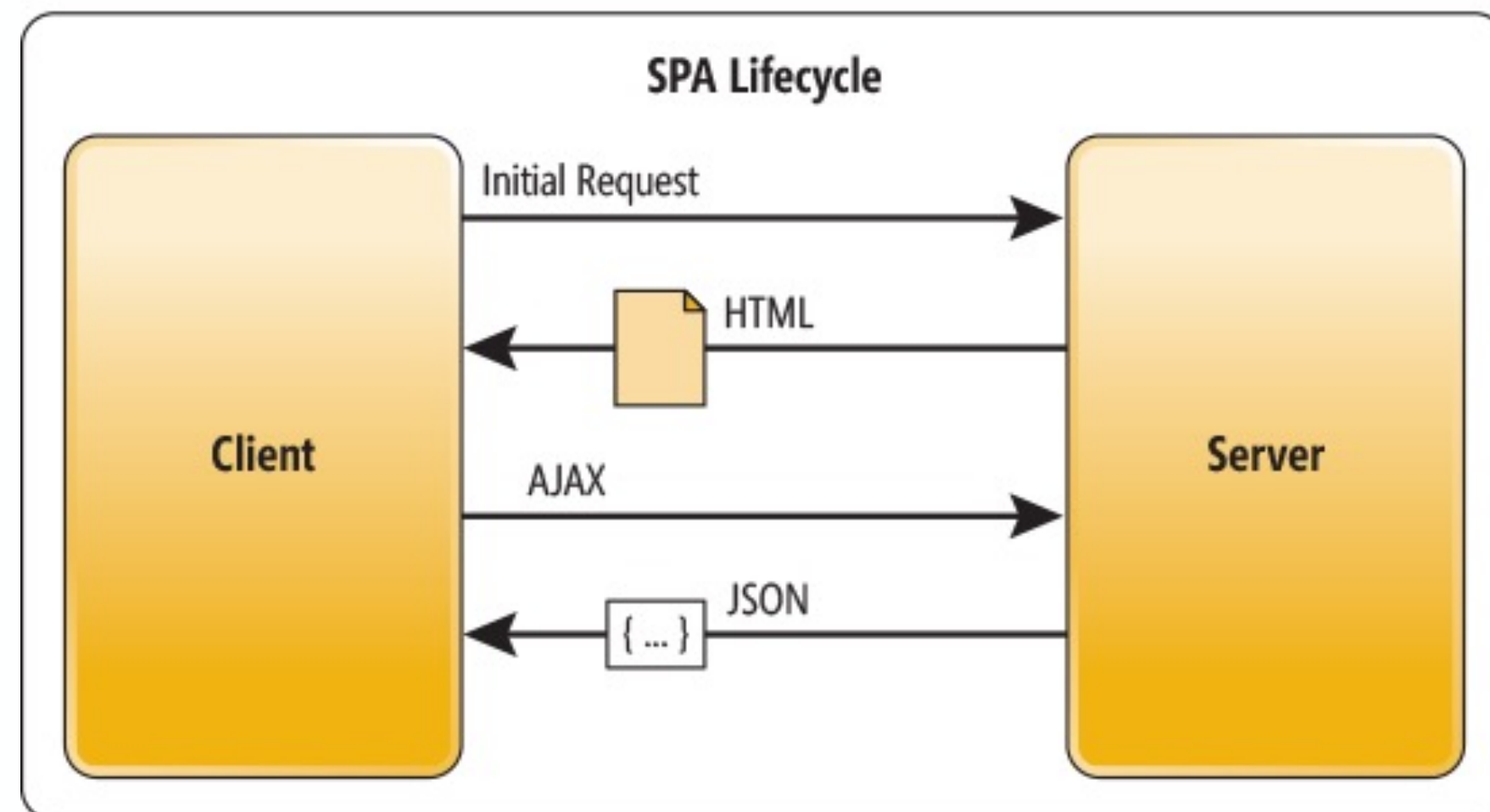


A multi-page web app is what we consider a traditional website.

Page reloads when moving from one page to another.

HTML and CSS is loaded at every page reload.

Example: Tiny App

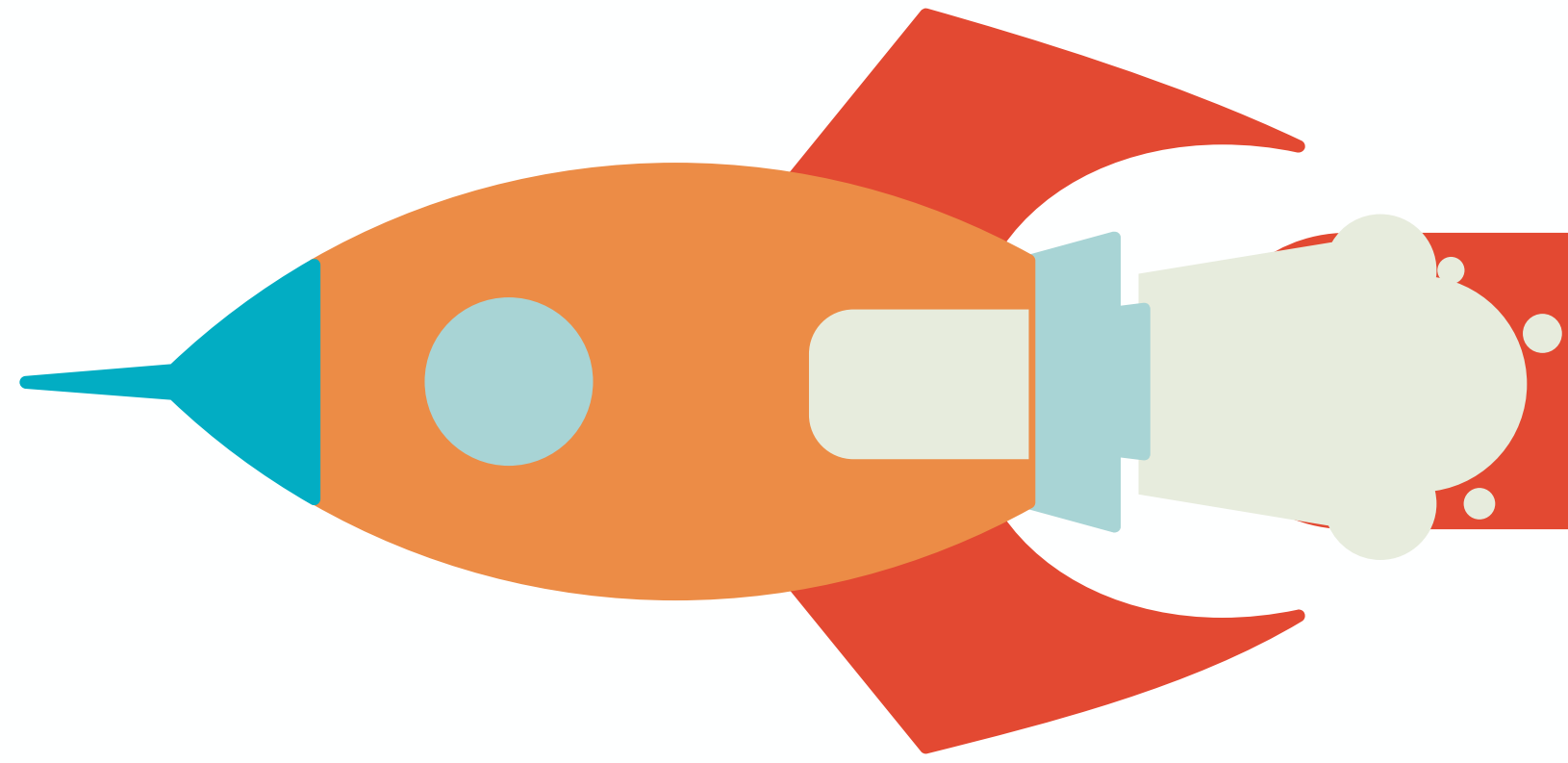


Content is loaded dynamically without needing to reload the page.

HTML and CSS is loaded once. Server sends back data (JSON).

Requests to the server are done with Ajax requests.

Example: Tweeter



Project Complete!

Dividing Tasks

Suggestion:

Pair program the first 3 features together with teammates.



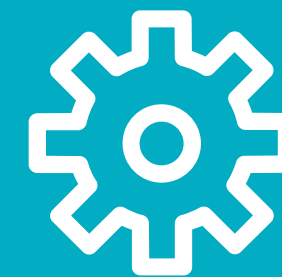
Project Setup

Follow Compass to setup project, GitHub and database.



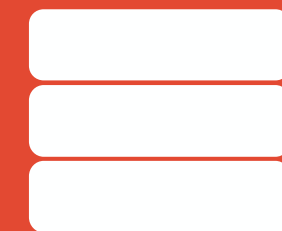
Frontend | Responsive

HTML, SCSS, Wireframes...etc.



Backend | API

Routes, Architecture, Logic...etc.

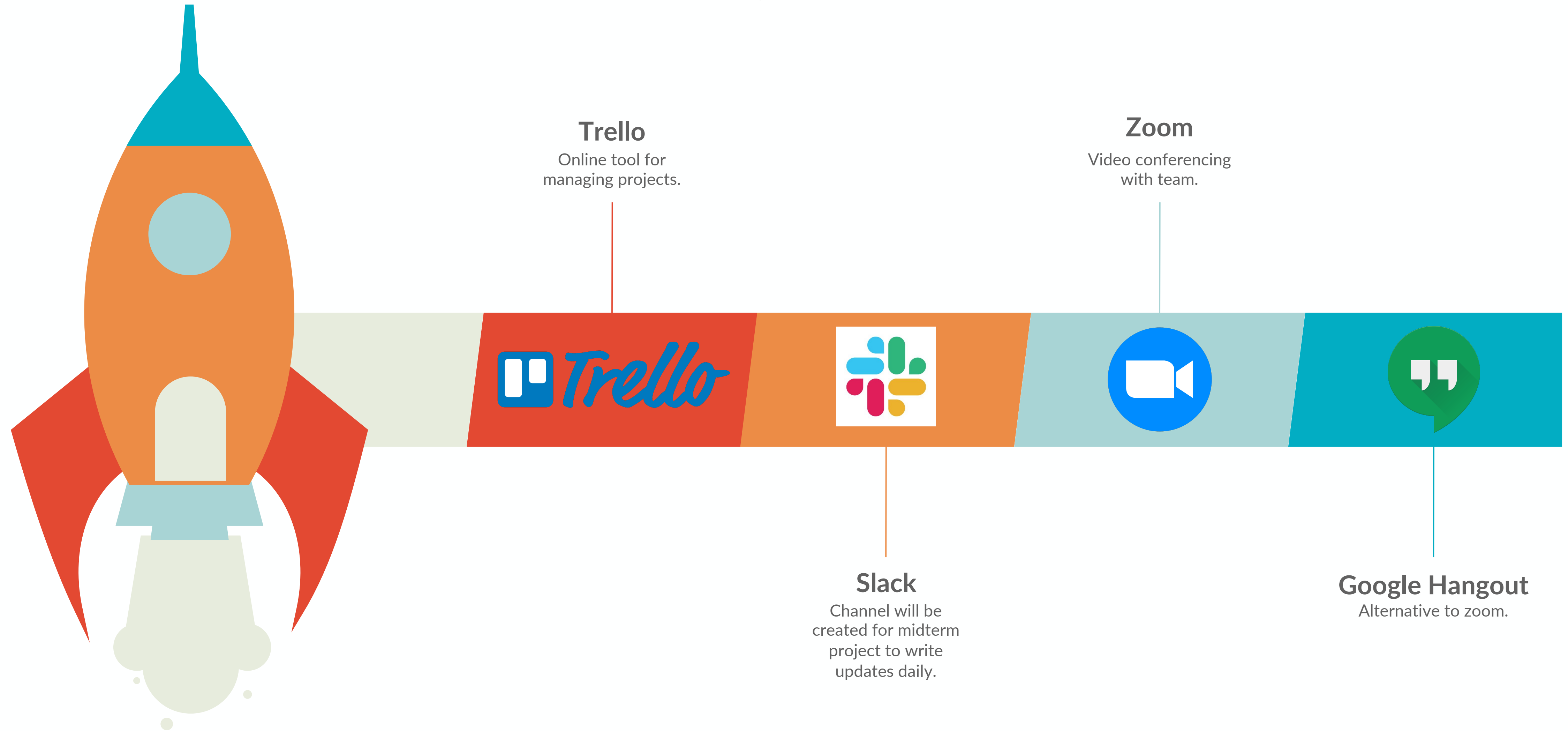


Database

ERD, Schemas, Seeds...etc.

Communication

Main Key to Success



Scrum Updates

Each day each team will post a scrum update in the midterm project channel in slack.

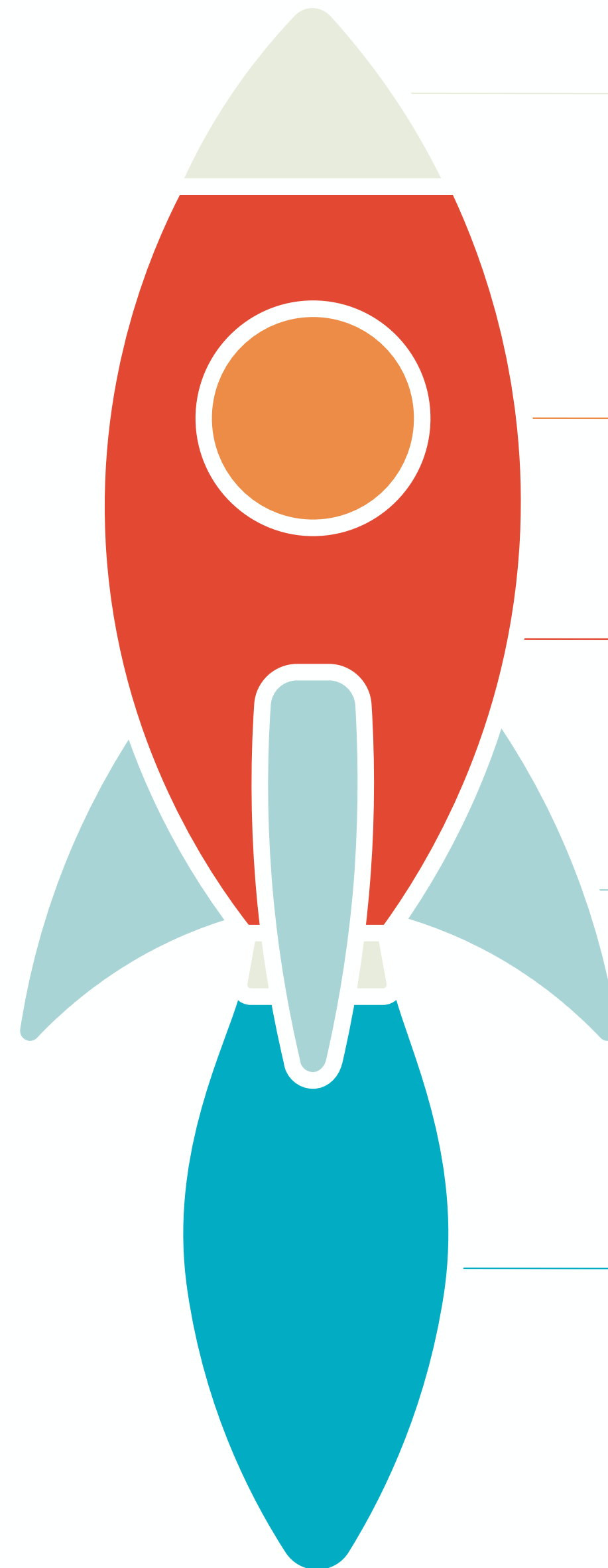
Y: Yesterday, I was working on...

T: Today, I will work on...

R: I'm stuck on ..., can someone pair program with me in your spare time?

W: I was able to refactor our routes.

P: I have 5/10 actions items completed (50%).



Yesterday

Describe what you worked on yesterday.

Today

Describe what you've been working on.

Roadblocks

Describe any challenges, if any.

Wins

Describe what you accomplished successfully.

Percentage

Estimate of how much you've completed.

Project Workflow

Project | GitHub | Database | Tools

Project Setup
Do the activities for database and git setup.
Clone the node skeleton.

01

Database Setup
Each team member will maintain own version
of the database.

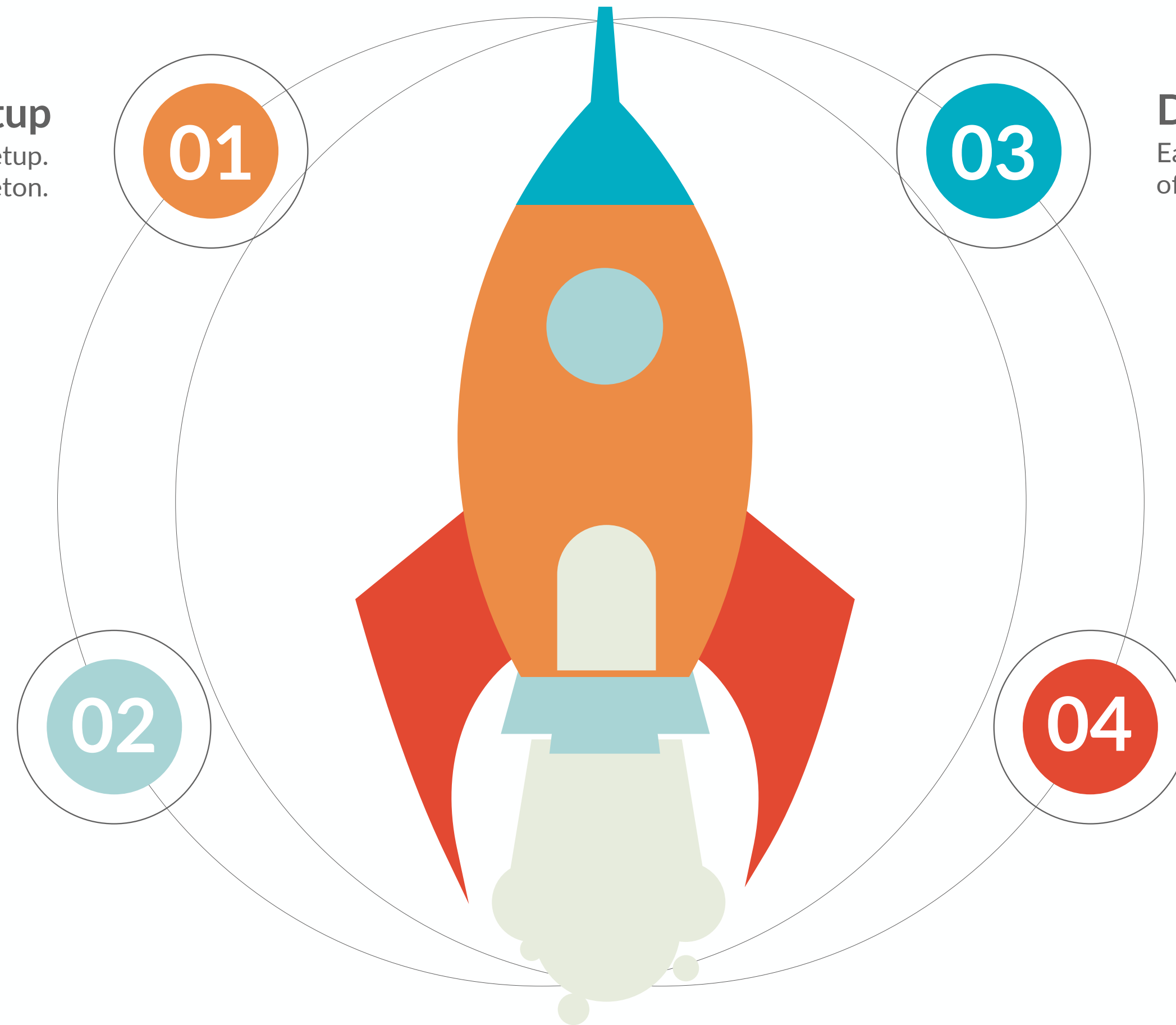
03

Git Workflow
NEVER CODE ON MASTER!!!

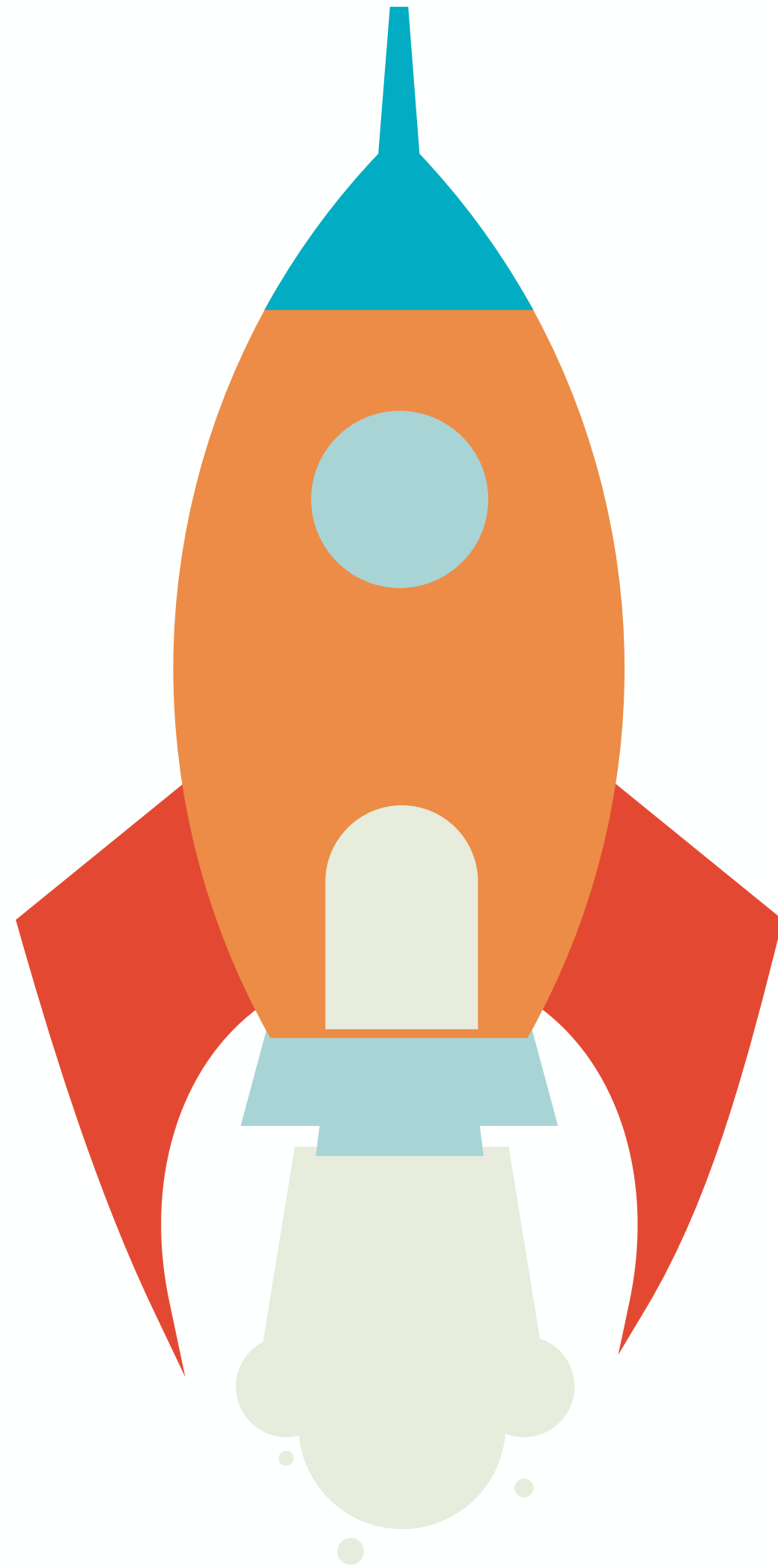
02

Tools
List of suggested tools in notes.

04







THANK YOU!

Instructor: Charlotte Lopez