

# Welcome!



# What Is Git?





# Git

The world's most popular  
**version control system**



# Version Control System?





# Version Control

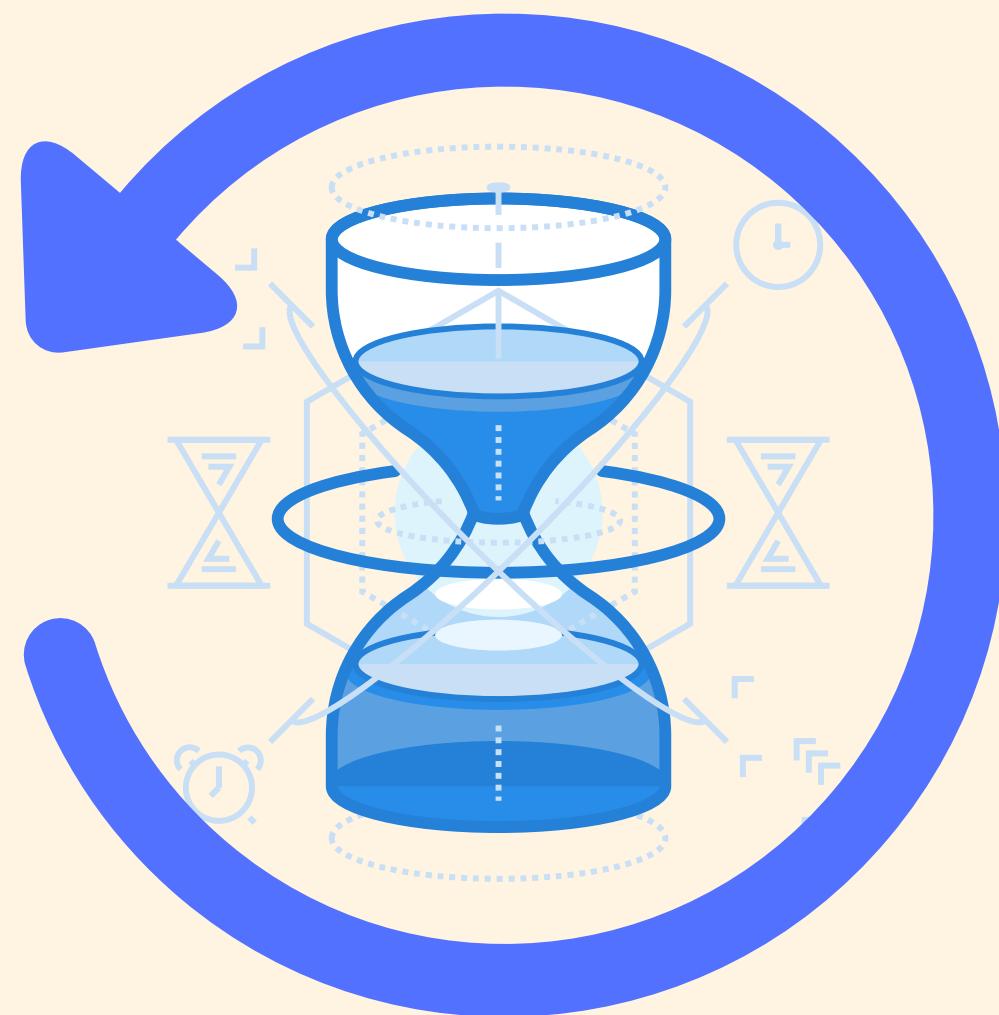
Version control is software that tracks and manages changes to files over time.



# Version Control

Version control systems generally allow users to revisit earlier versions of the files, compare changes between versions, undo changes, and a whole lot more.

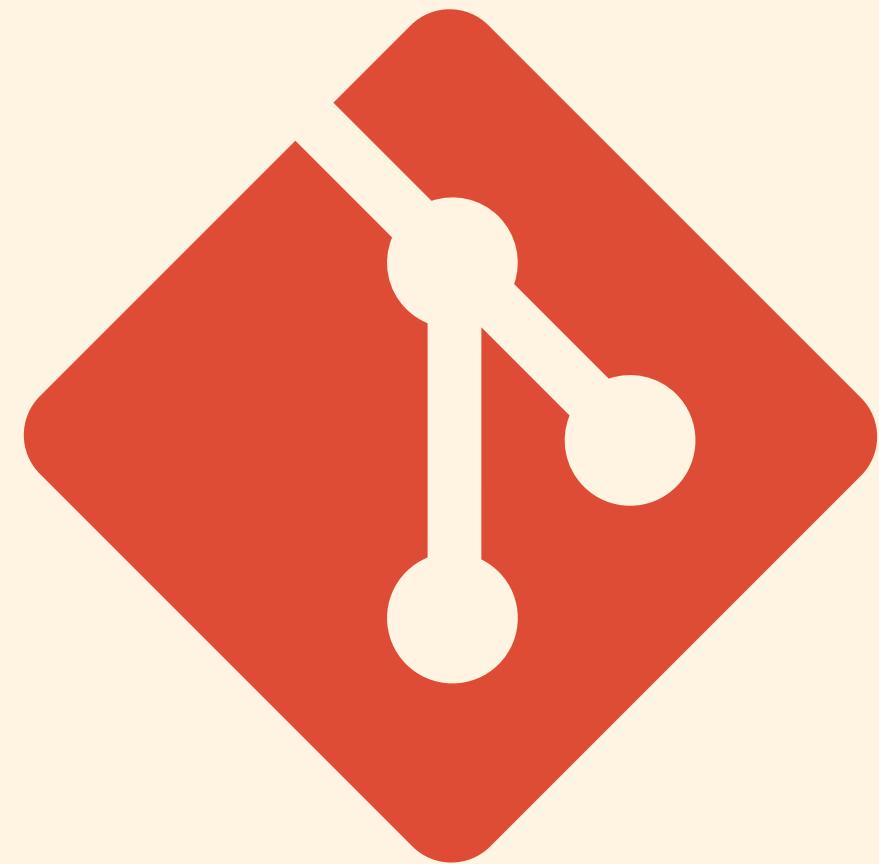
\* We'll go into more detail shortly!

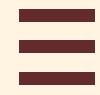


# Git is just one VCS

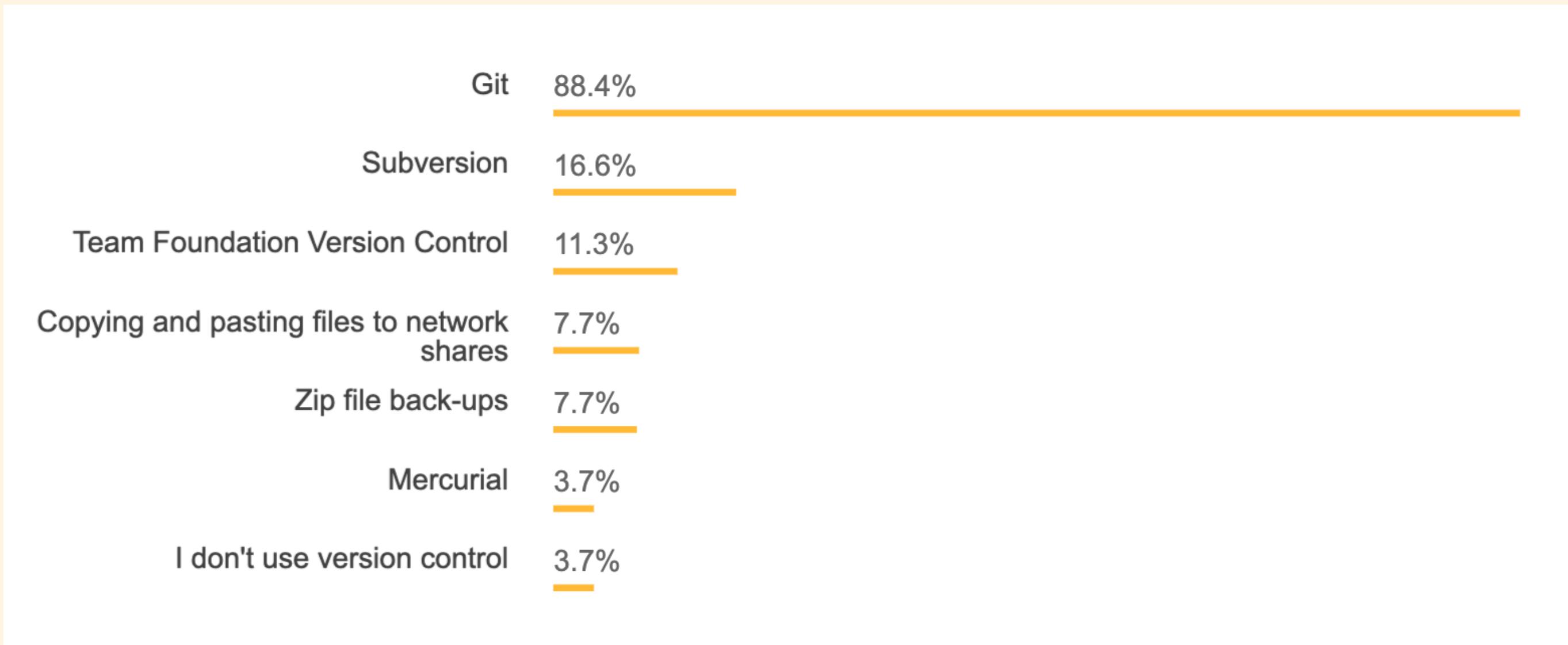
Git is just one of the many version control systems available today. Other well-known ones include Subversion, CVS, and Mercurial.

They all have similar goals, but vary significantly in how they achieve those goals. Fortunately, we only need to care about Git because...





# Git is the clear "winner"



In Stack Overflow's 2018 Developer Survey, nearly 90% of respondents reported Git as their version control system of choice. Over the last few years, the survey hasn't even bothered to ask about version control because Git is so widely used.





Life Without  
Git



# Git-ing To The Point

What exactly does Git do for us?



# Git helps us...

Track changes across multiple files

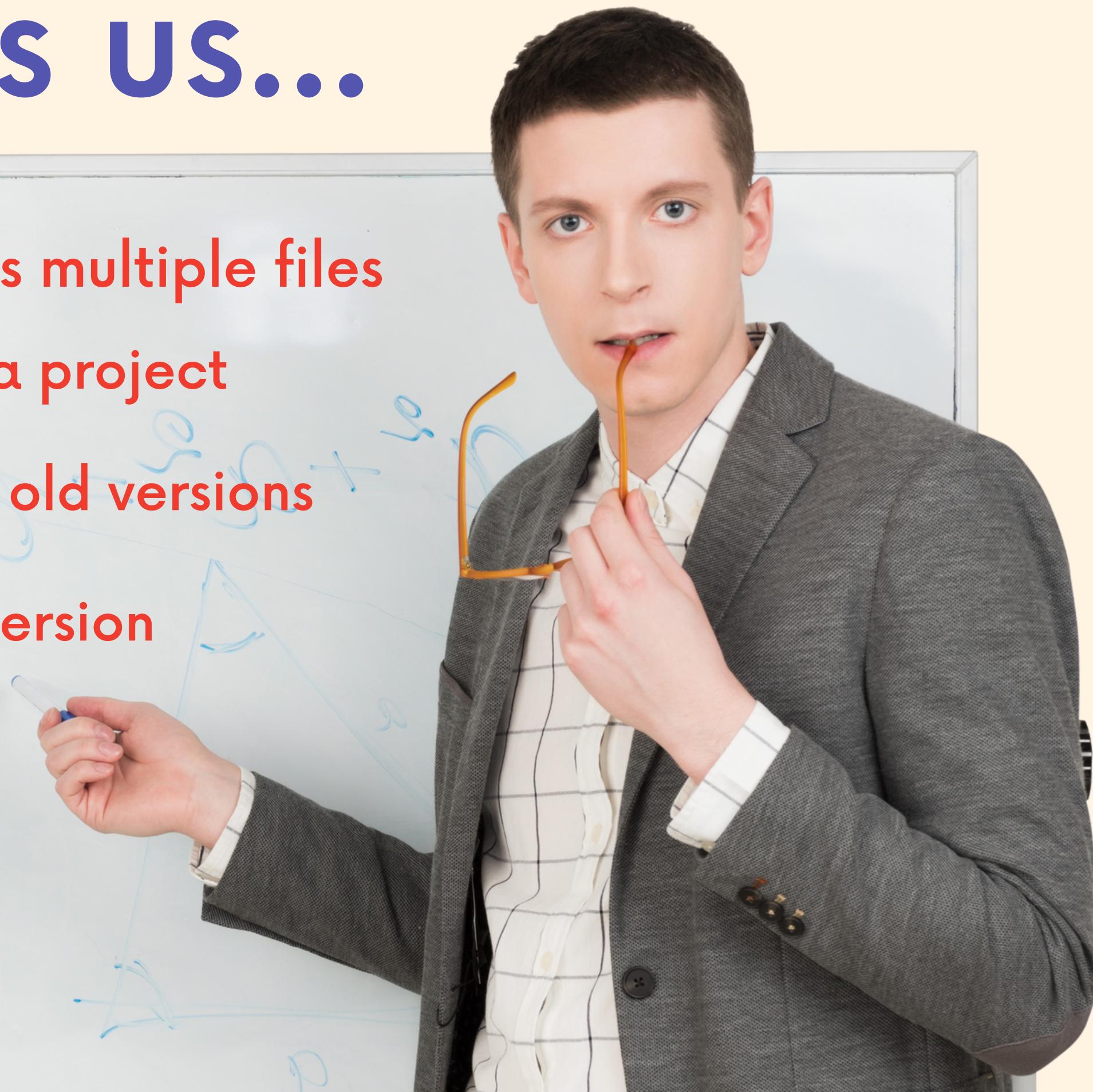
Compare versions of a project

"Time travel" back to old versions

Revert to a previous version

Collaborate and  
share changes

Combine changes



# The Big Picture



Oh boy, I sure do  
love playing my  
video games!



I'm going to save  
my game now in  
case I die soon!



Oh jeez, this is  
going to be a  
difficult fight!



ughhhh I died!





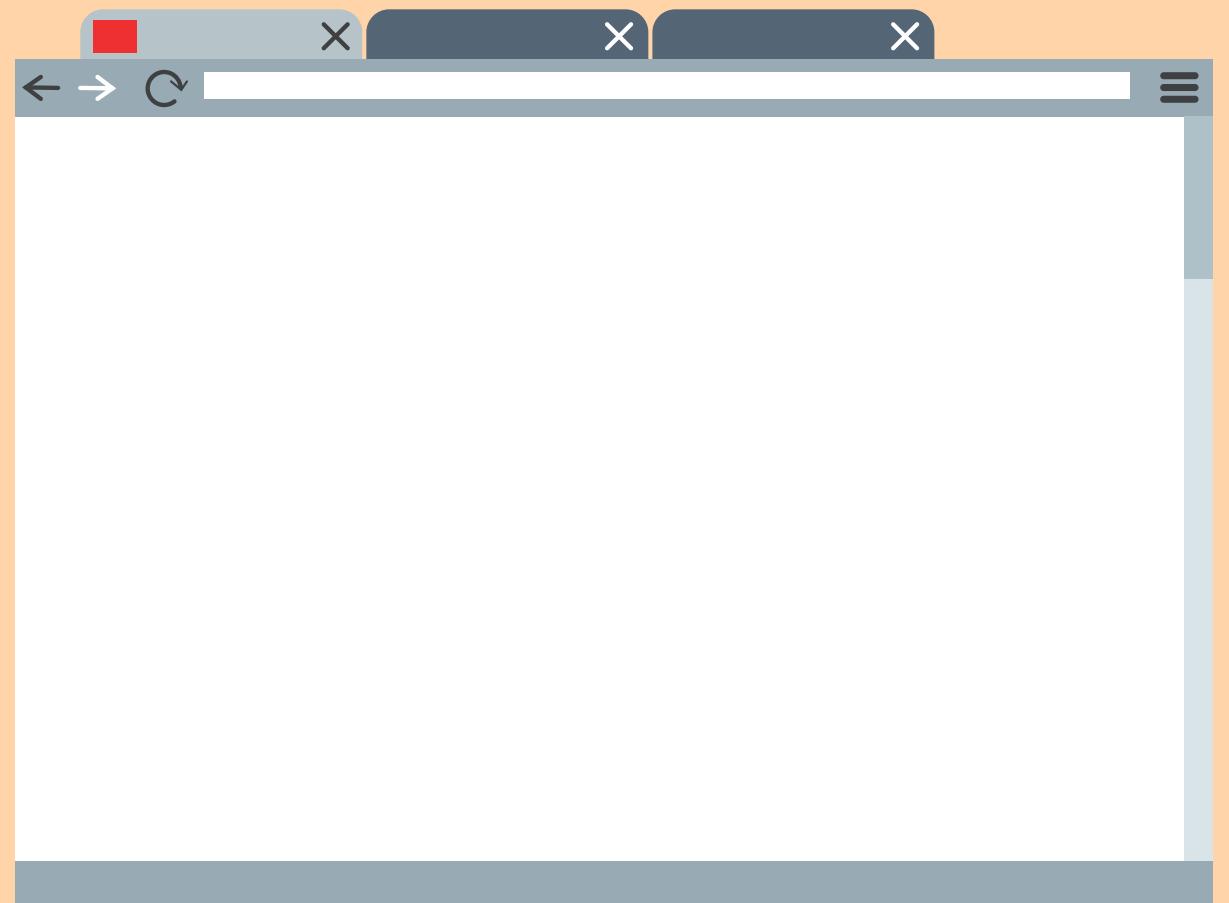
Thank heavens I  
saved my game!  
I can just revert!



# Let's Look At An Example

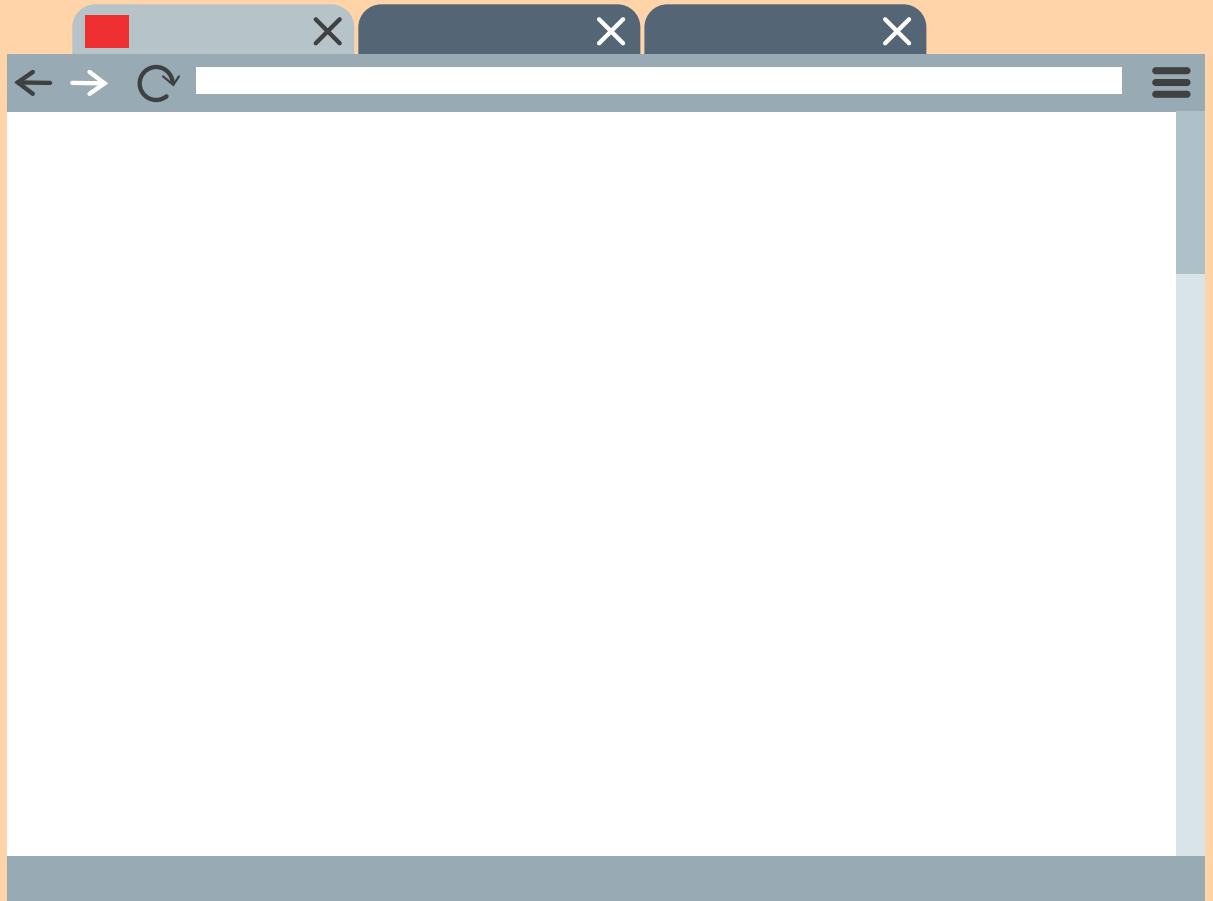
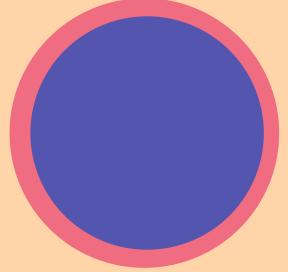


# I Start A New Project!



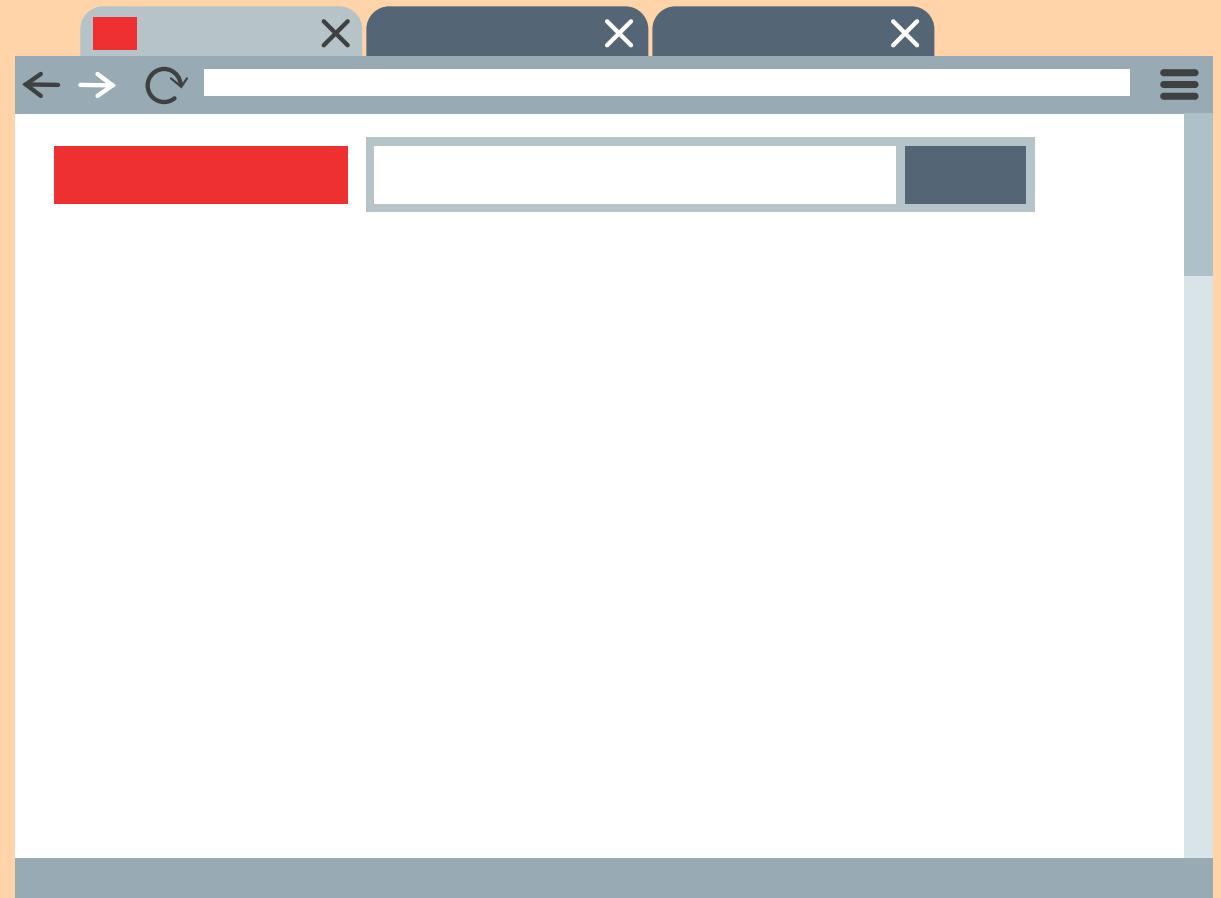
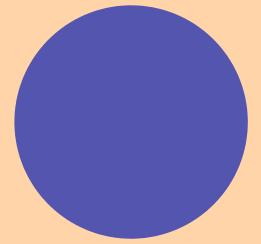
# Add A Checkpoint

Initialize Project



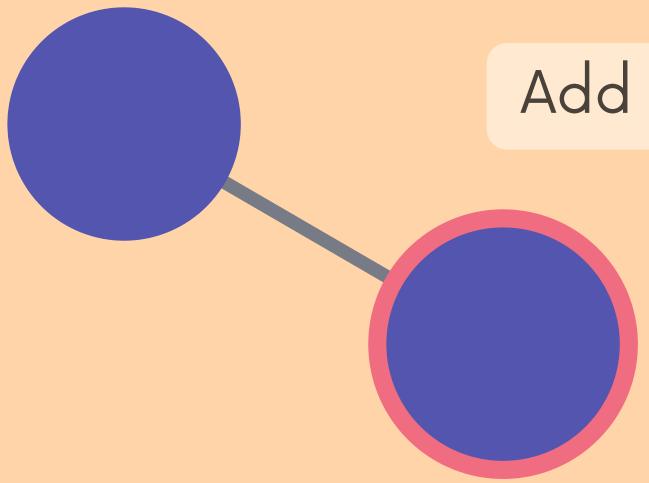
# I work on the navbar

Initialize Project

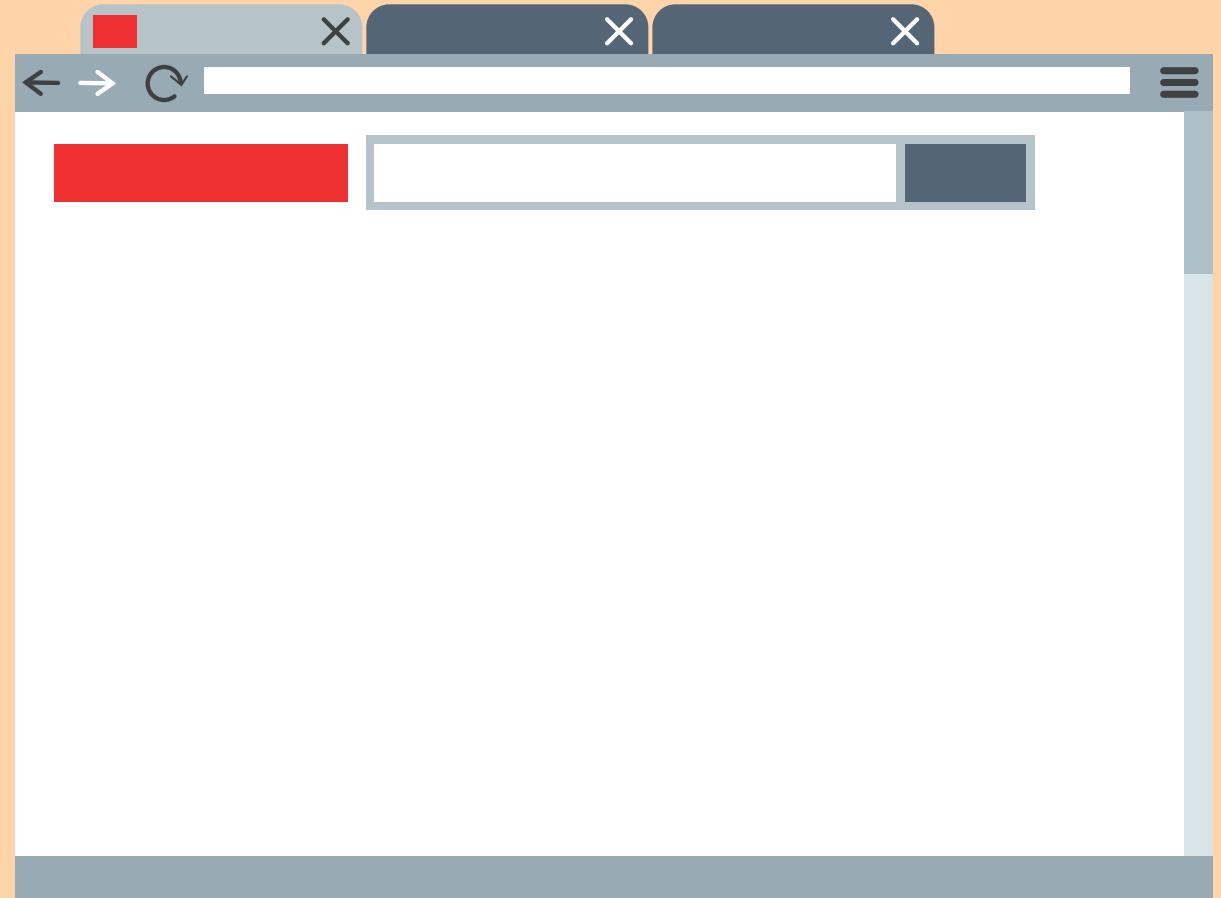


# Add A Checkpoint

Initialize Project



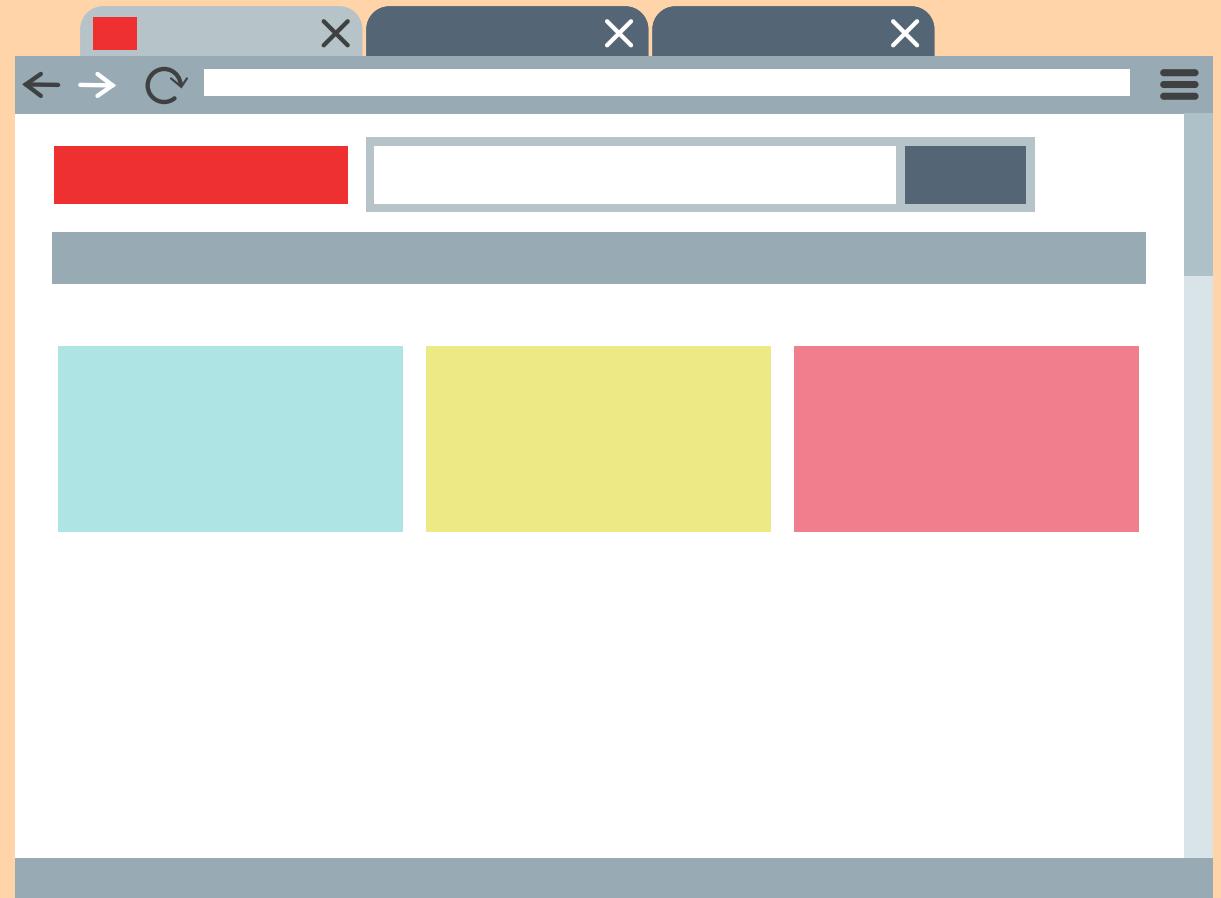
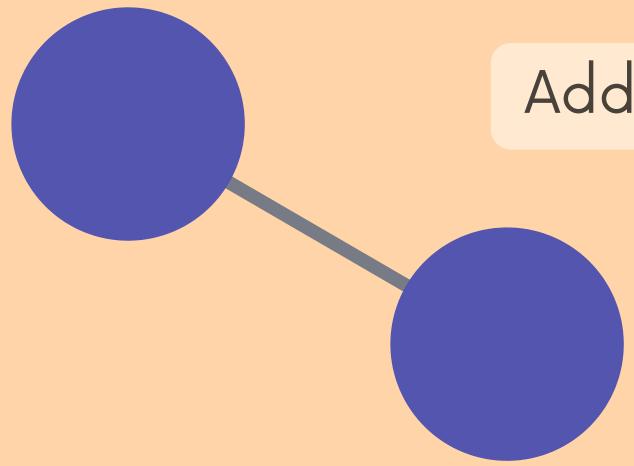
Add Top Navbar



# I add some content

Initialize Project

Add Top Navbar

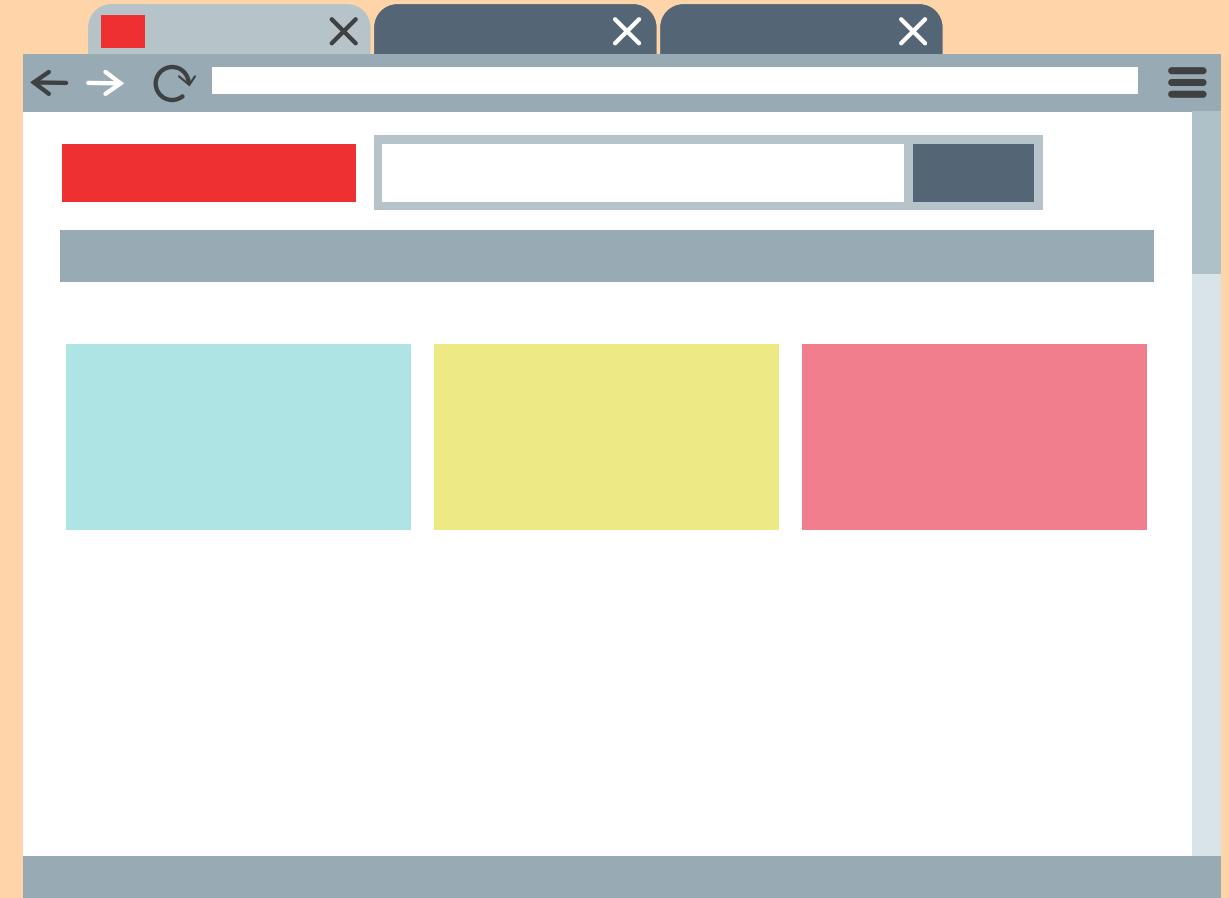


# Add A Checkpoint

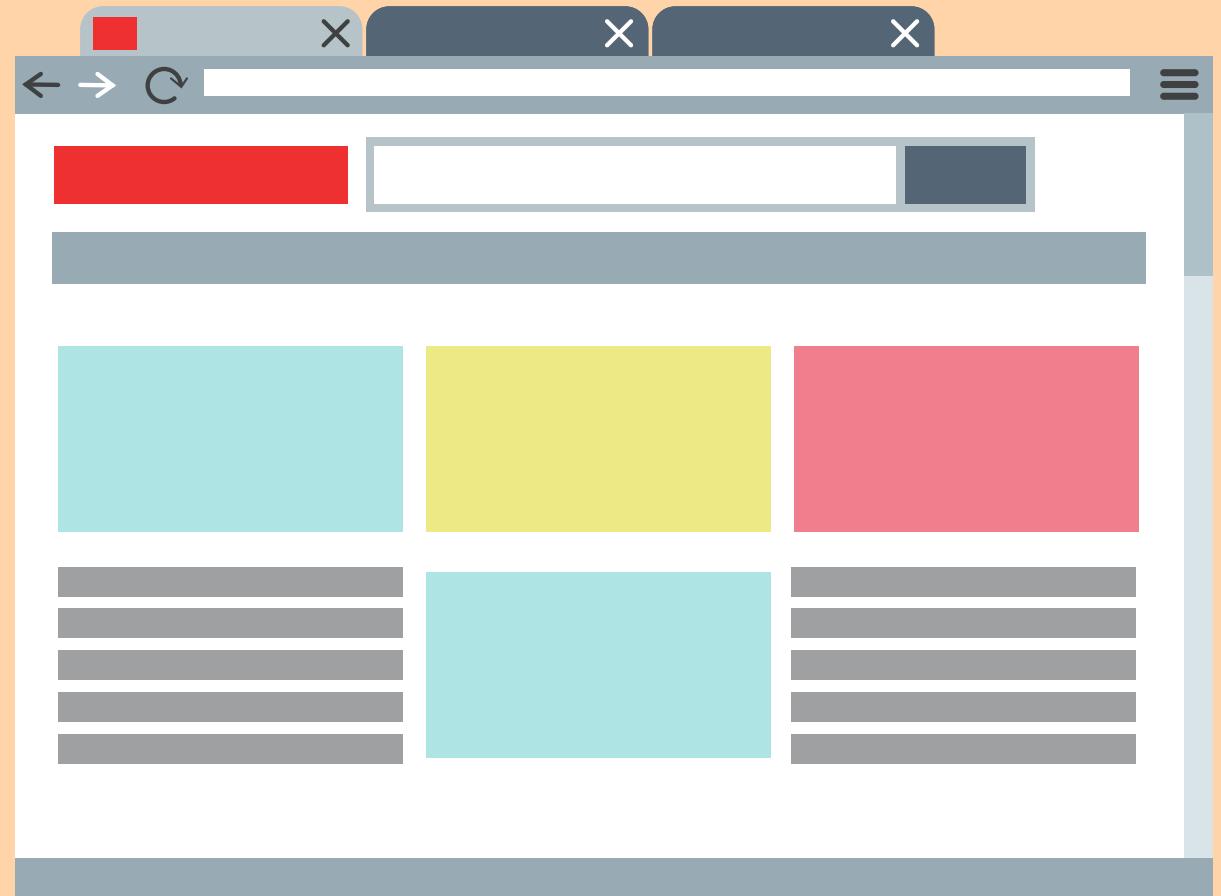
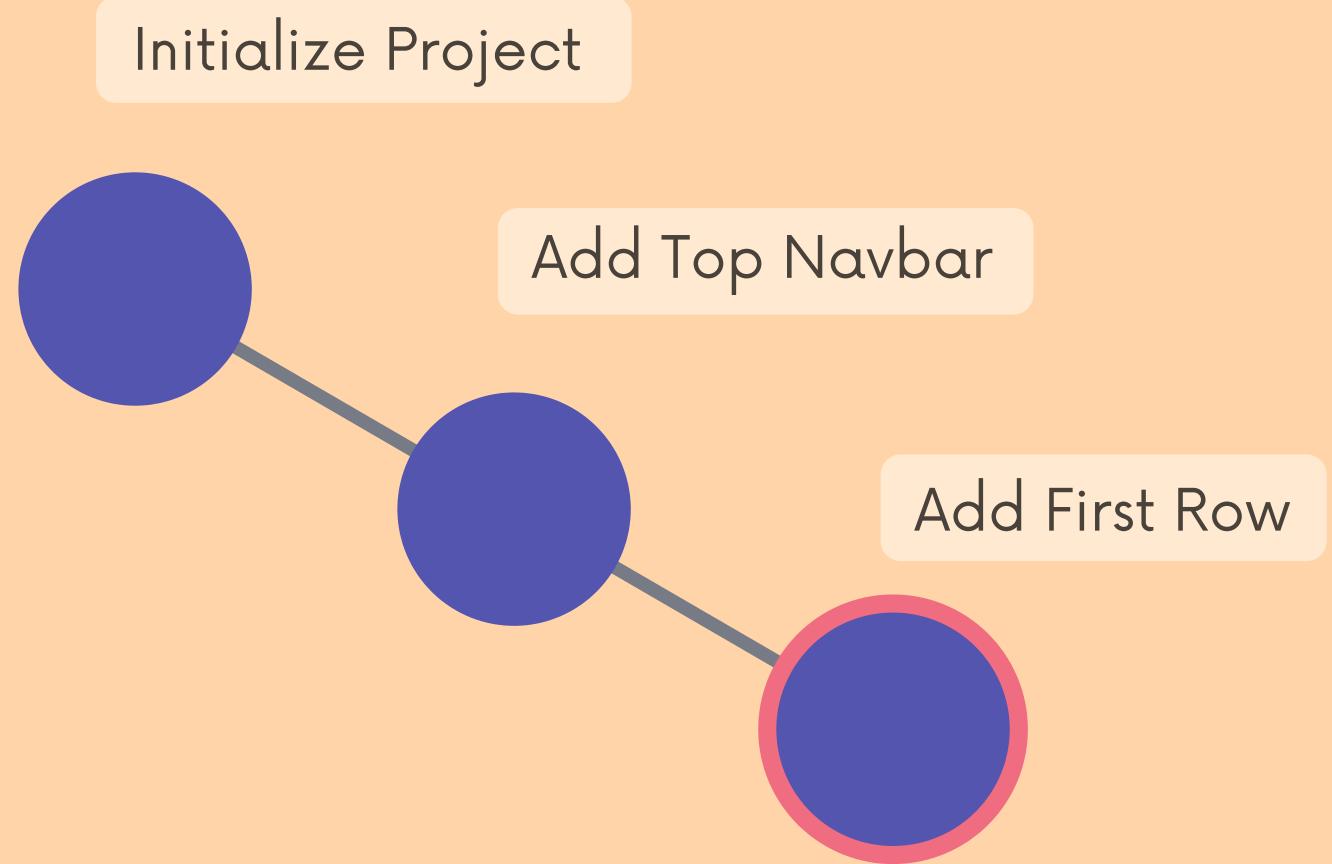
Initialize Project

Add Top Navbar

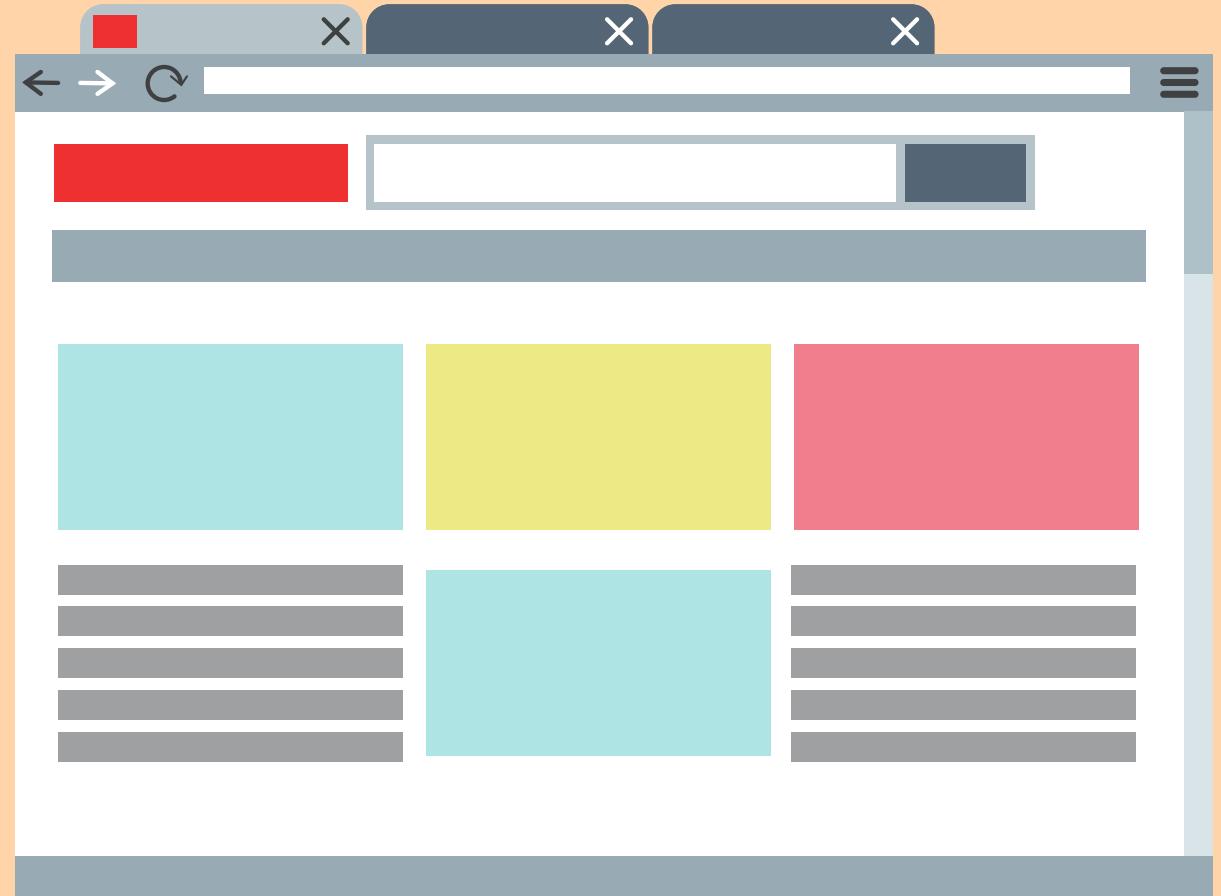
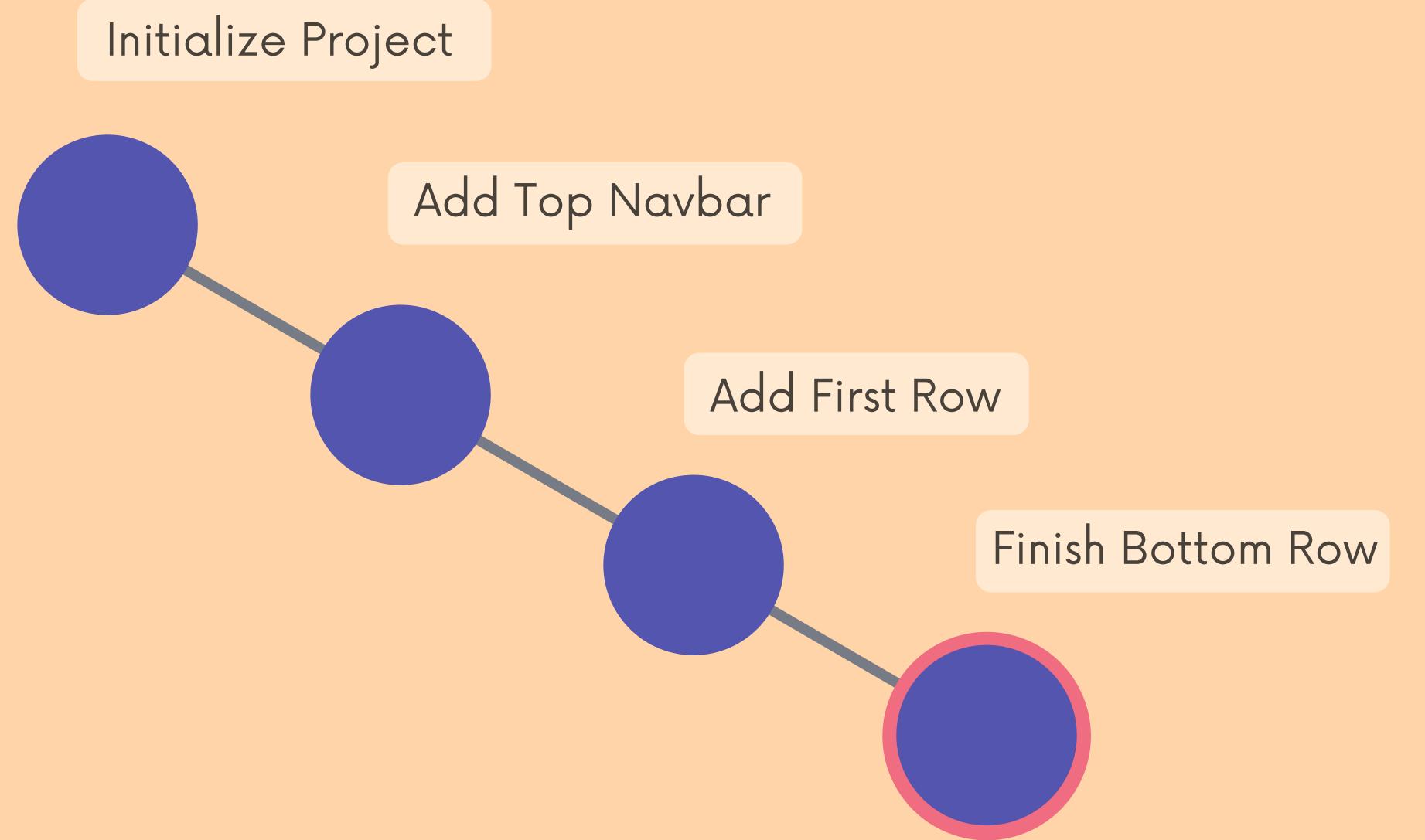
Add First Row



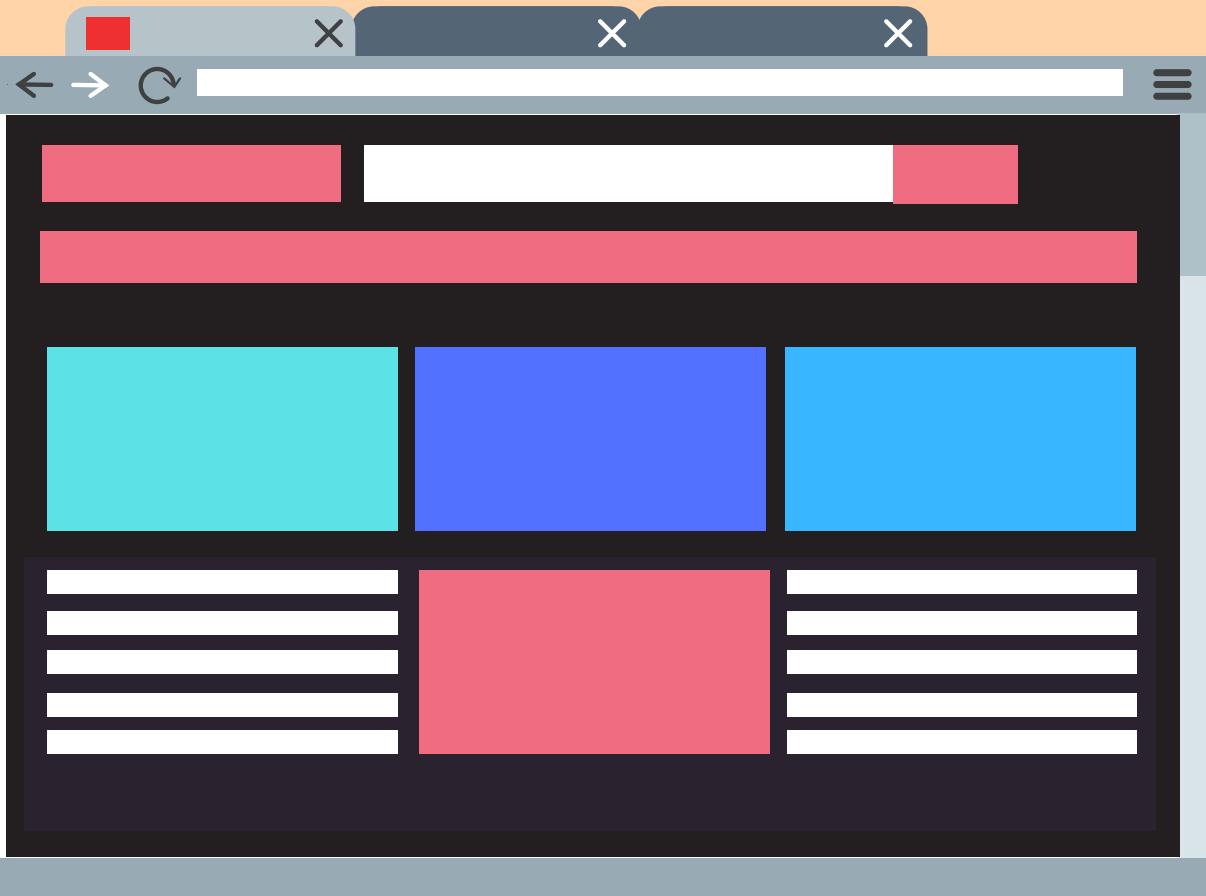
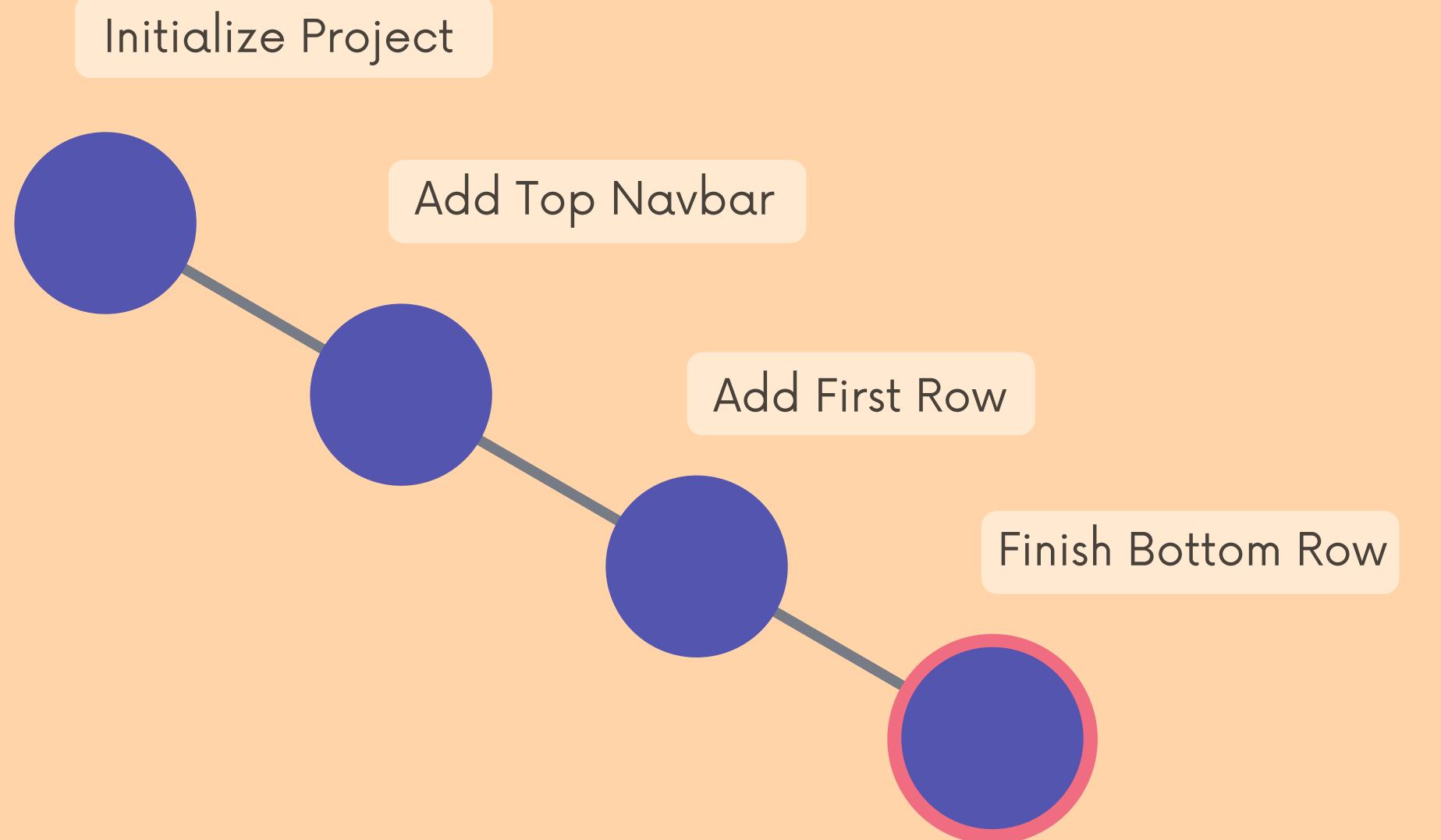
# I add more content



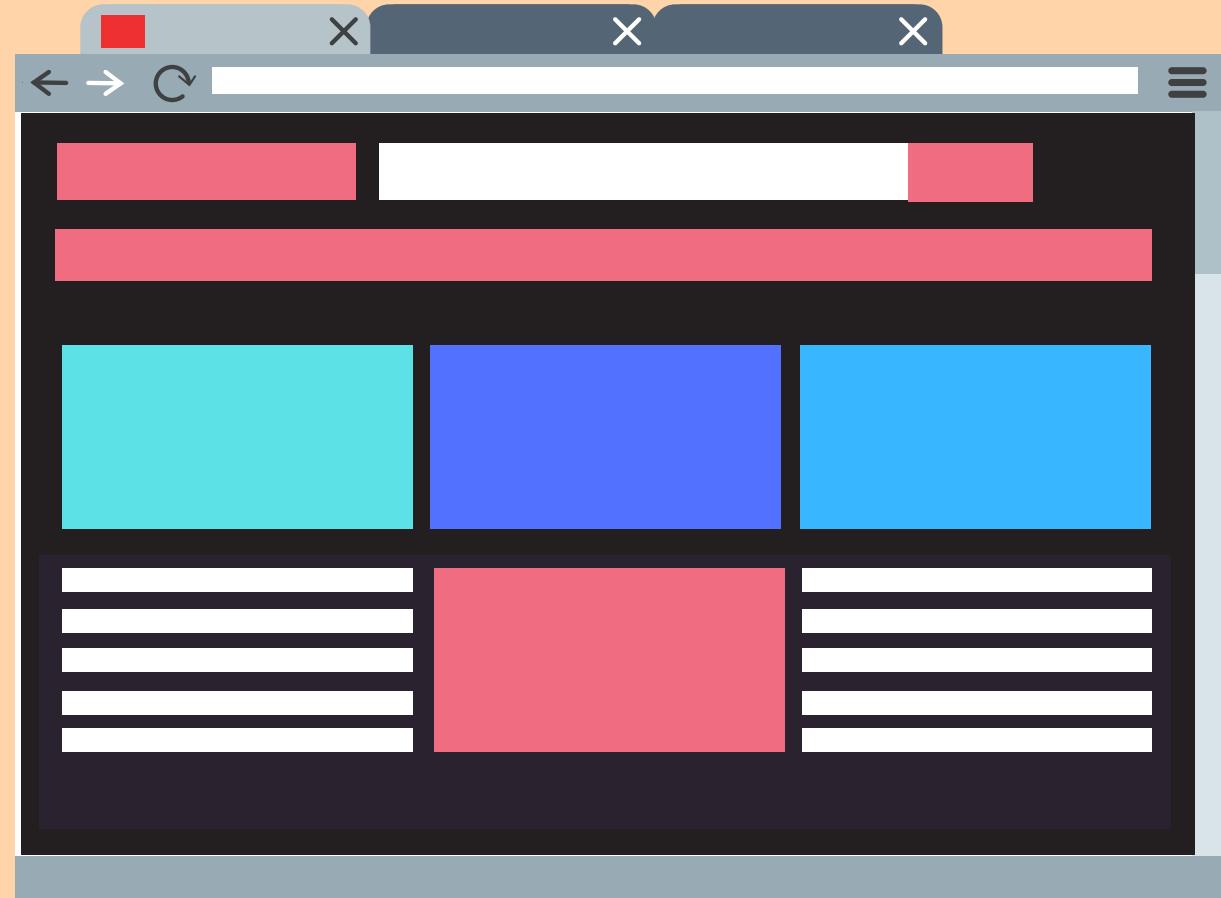
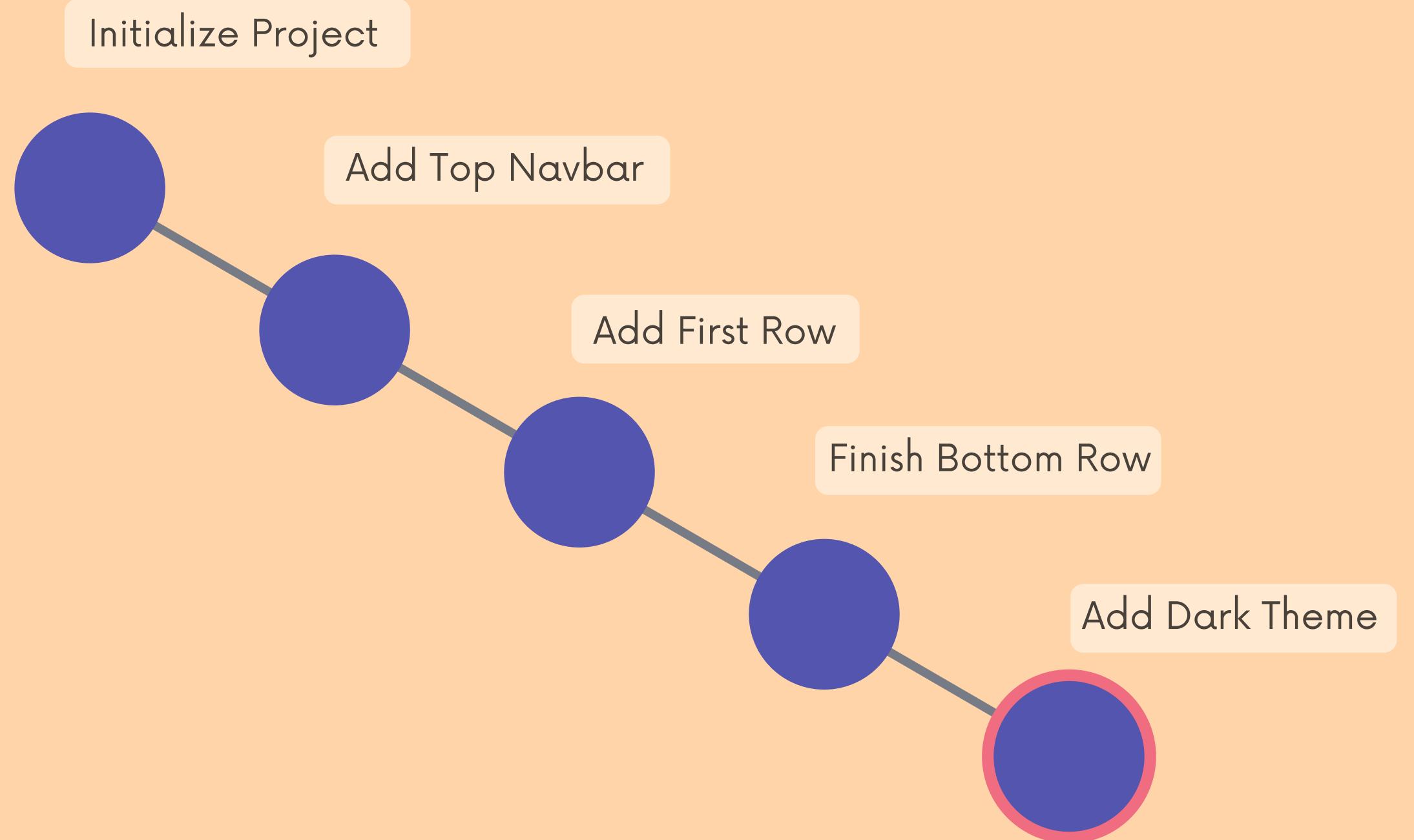
# Add A Checkpoint



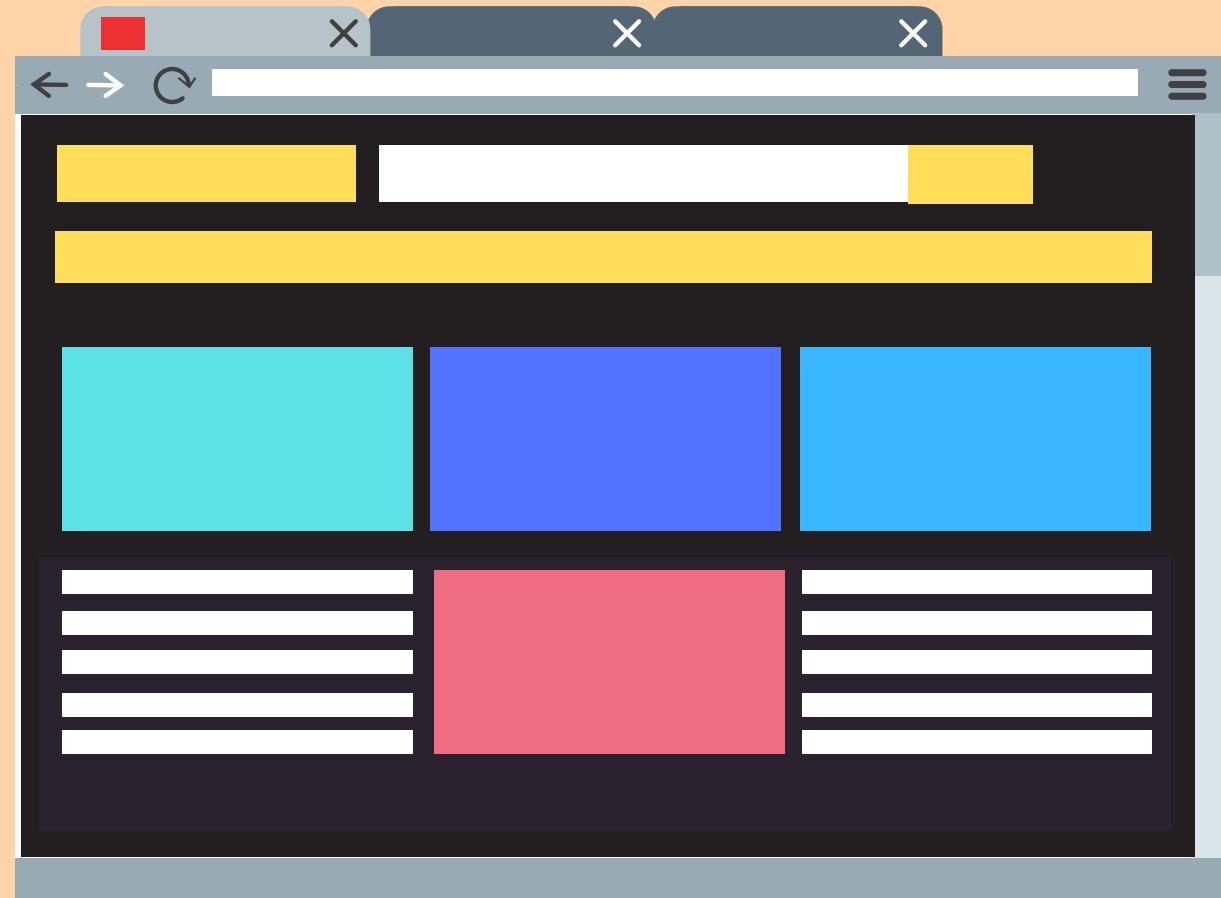
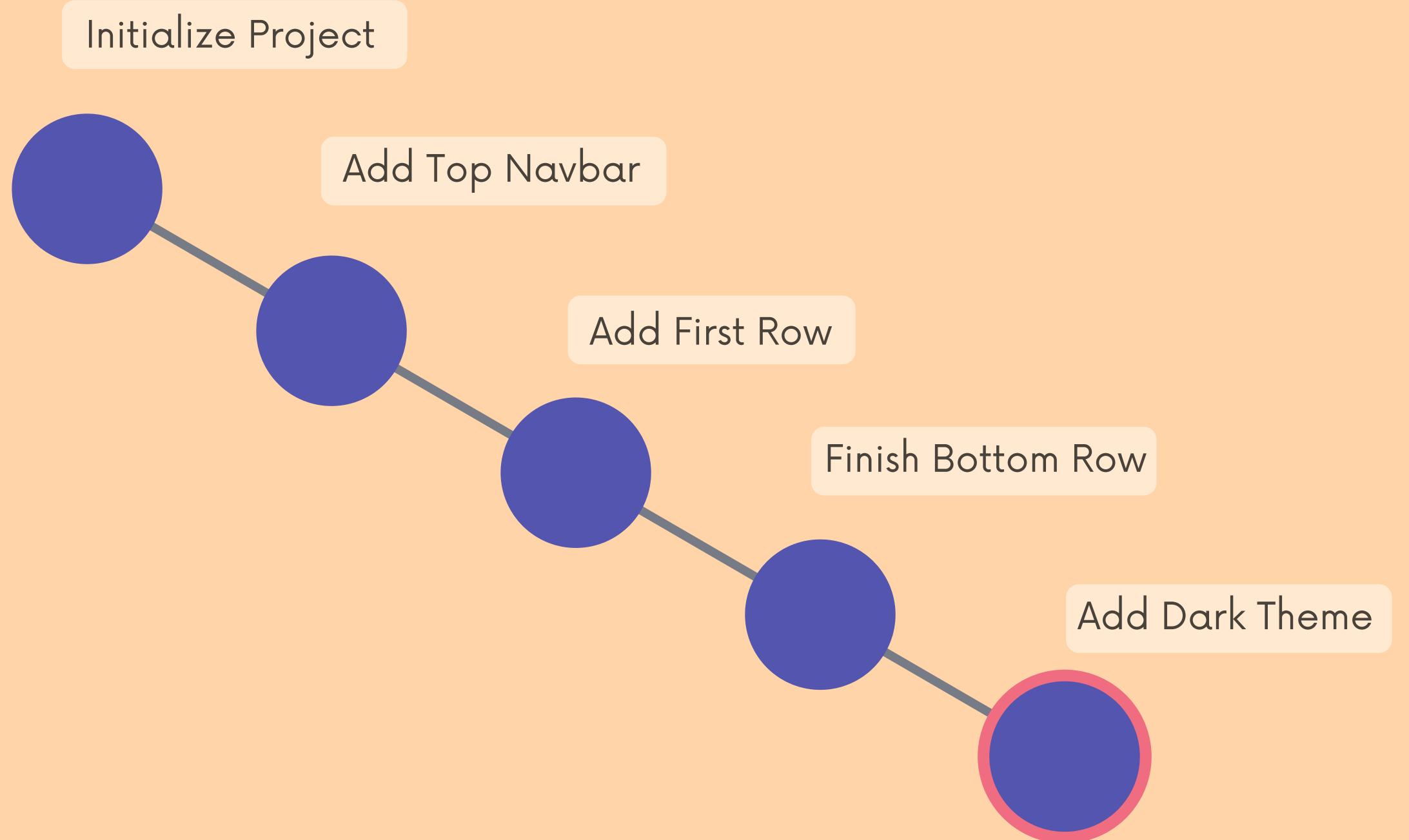
# I change the theming



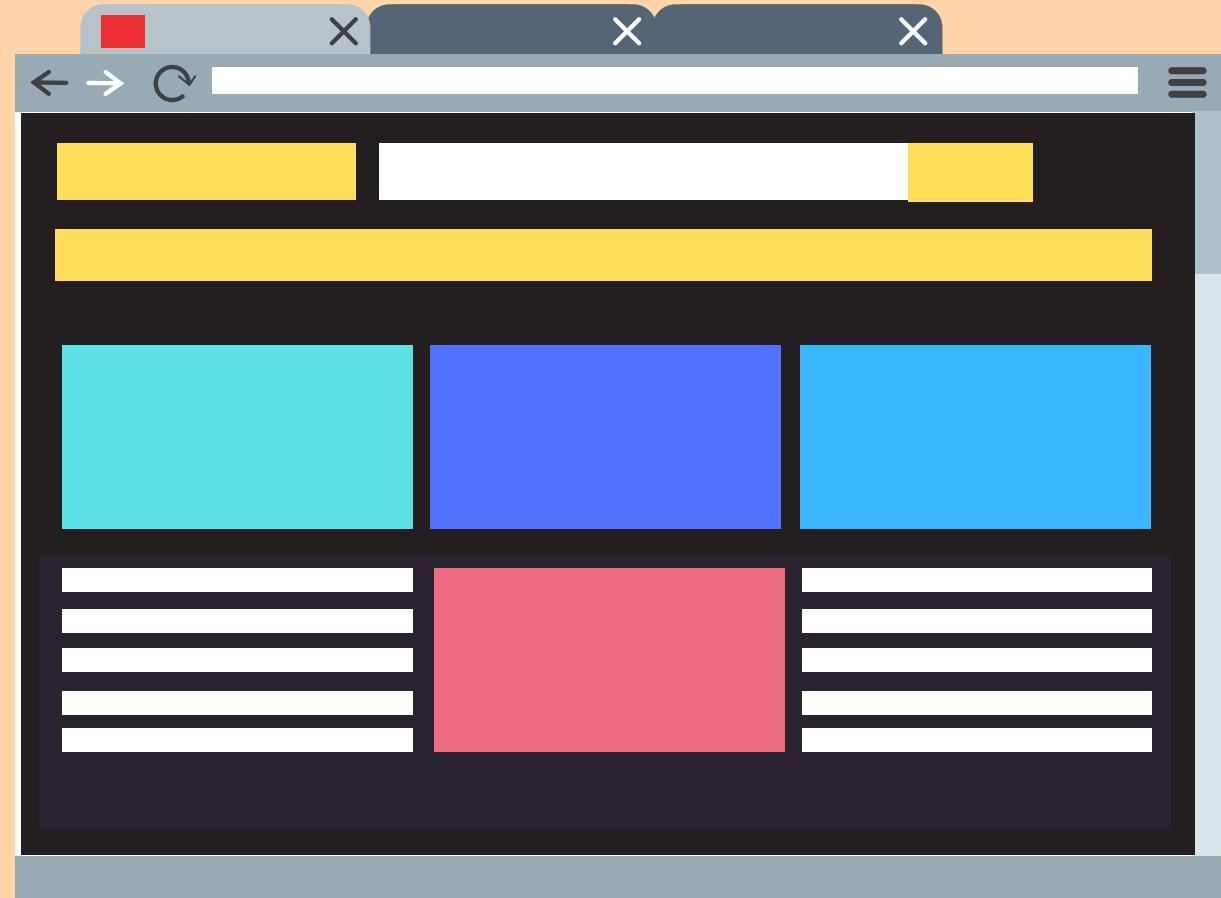
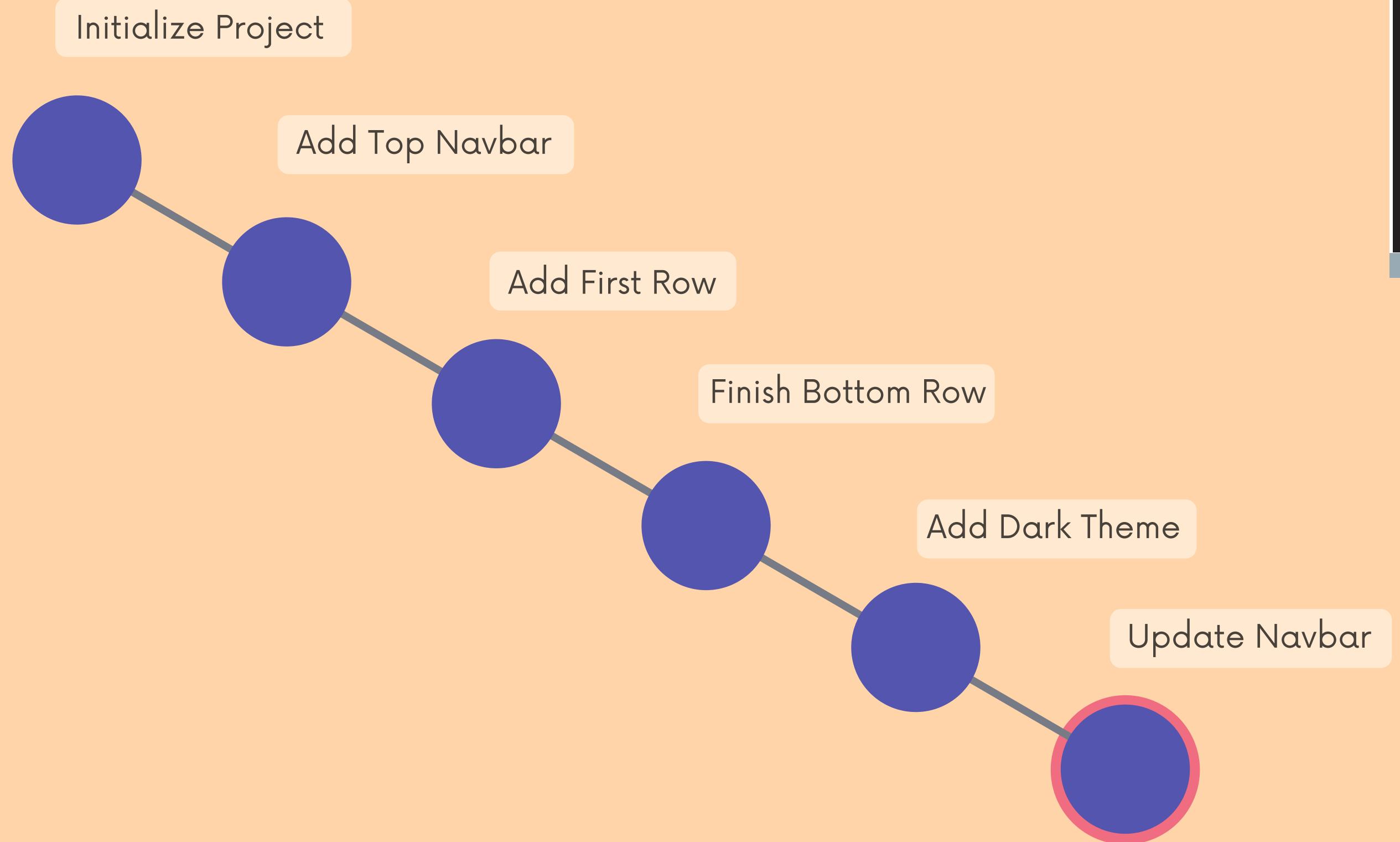
# Add A Checkpoint



# I alter the navbar



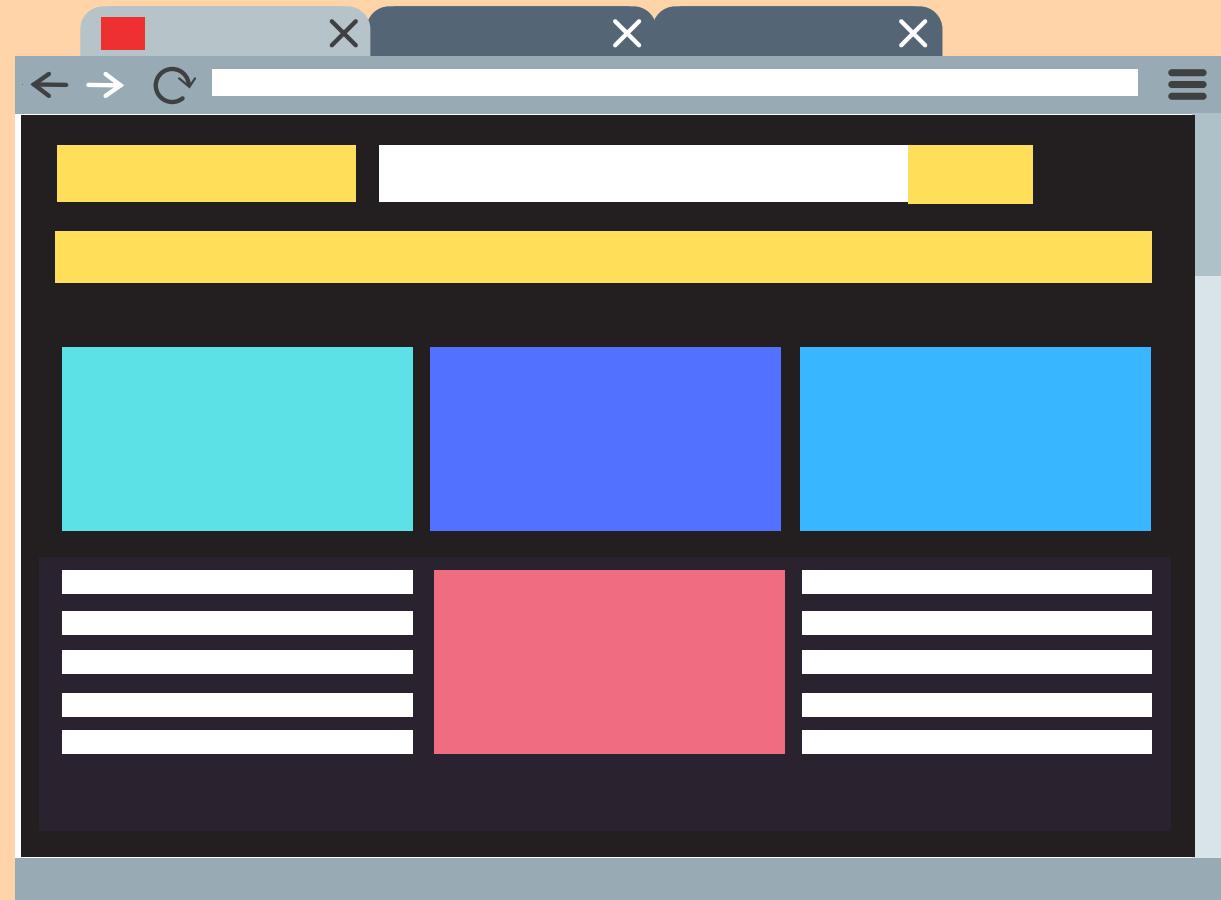
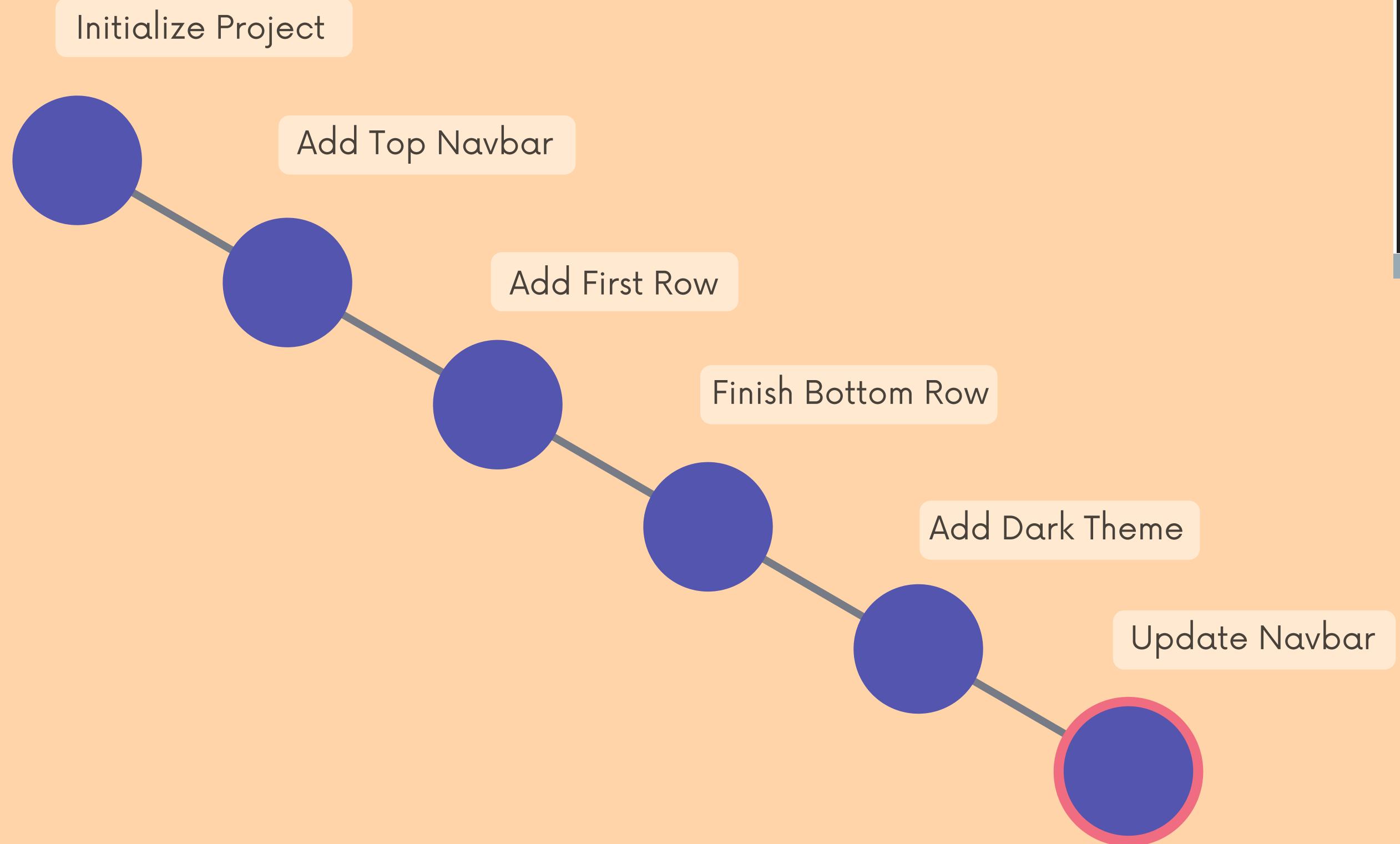
# Add A Checkpoint



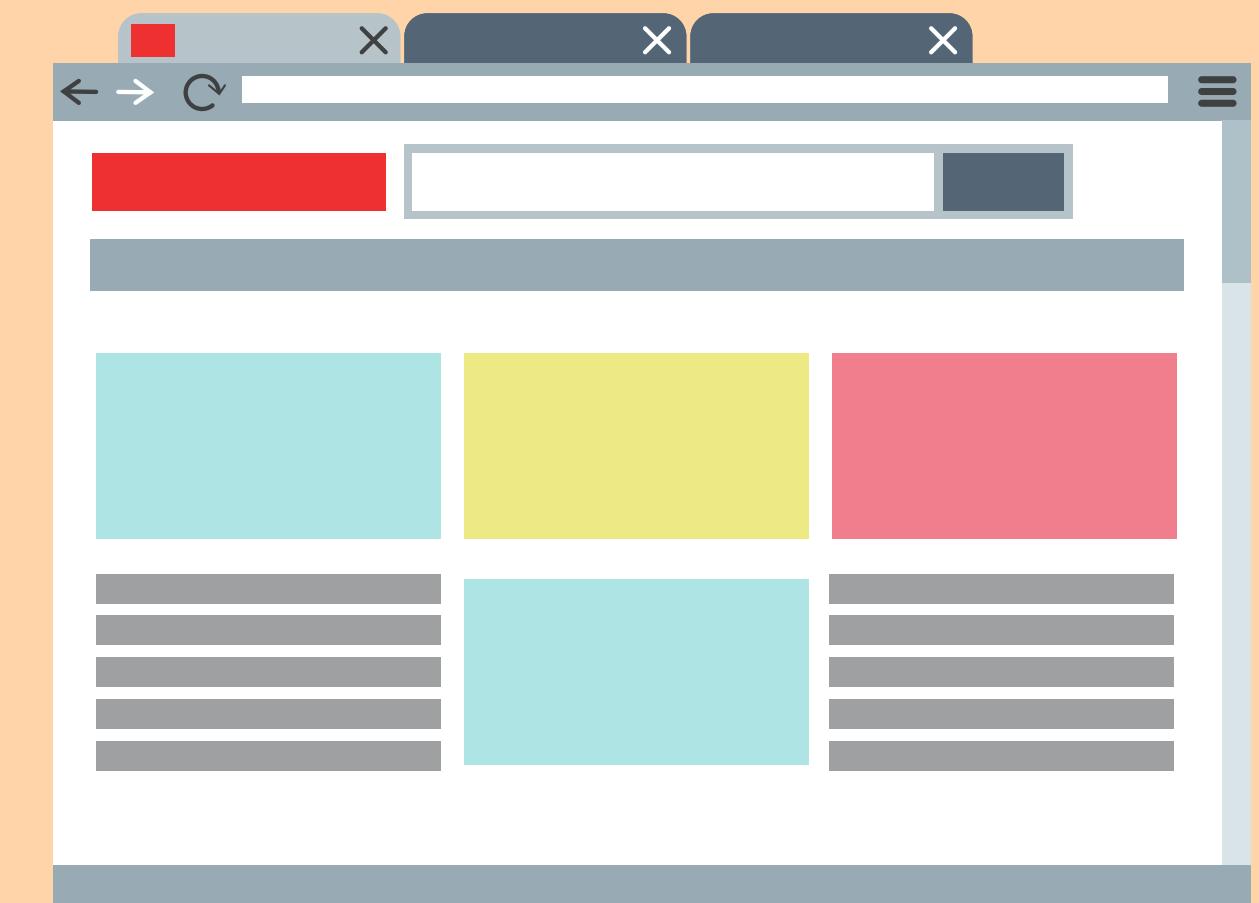
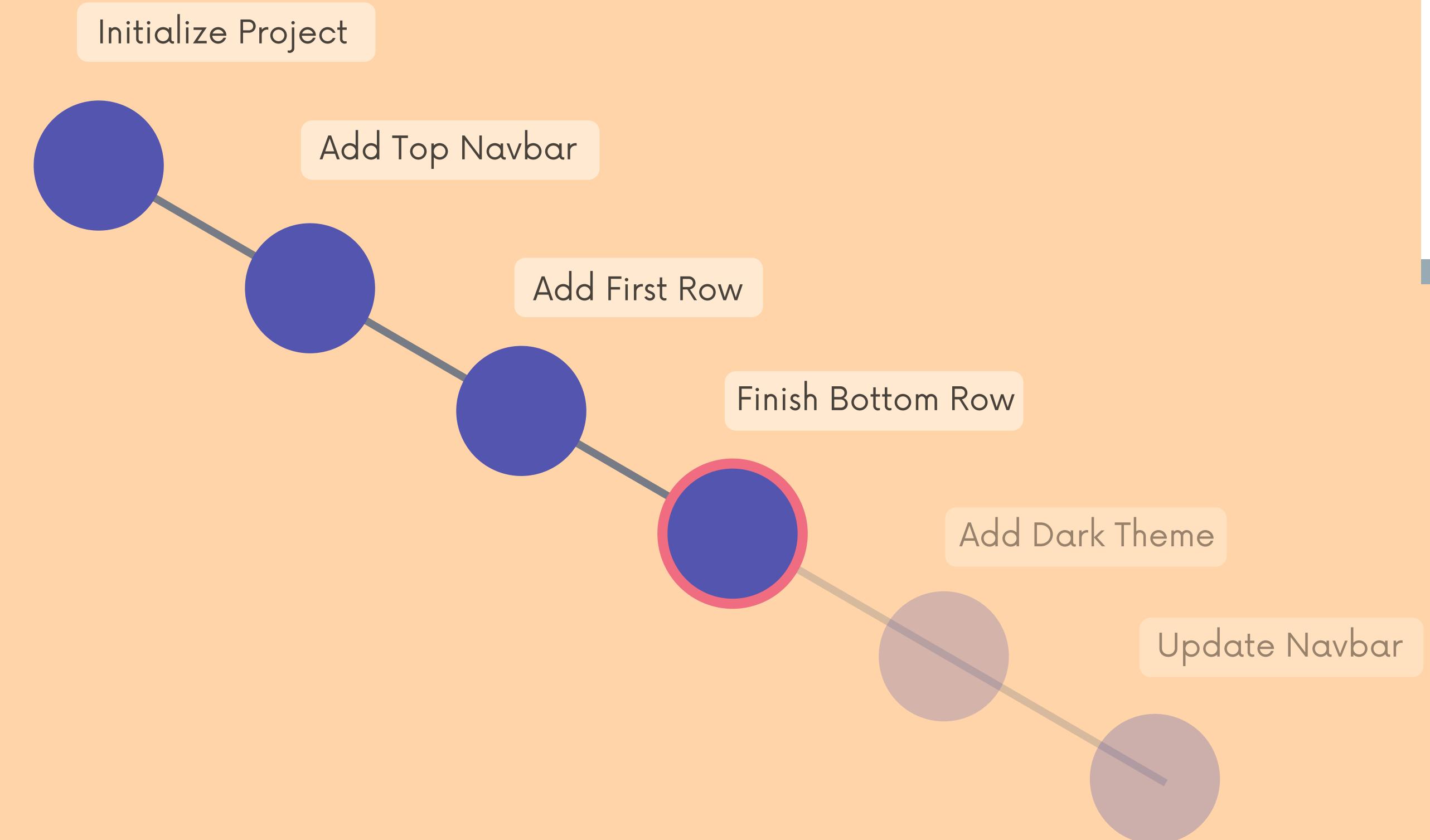
A woman with long dark hair, wearing a brown herringbone blazer over a light-colored button-down shirt and khaki shorts, is captured in a dynamic pose. She is leaning forward with her right arm extended, pointing her index finger towards the right side of the frame. Her mouth is open as if she is shouting or speaking with intensity. The background is a solid, warm orange color.

**ANGRY BOSS SAYS...**  
**THE COLORS ARE BAD!**

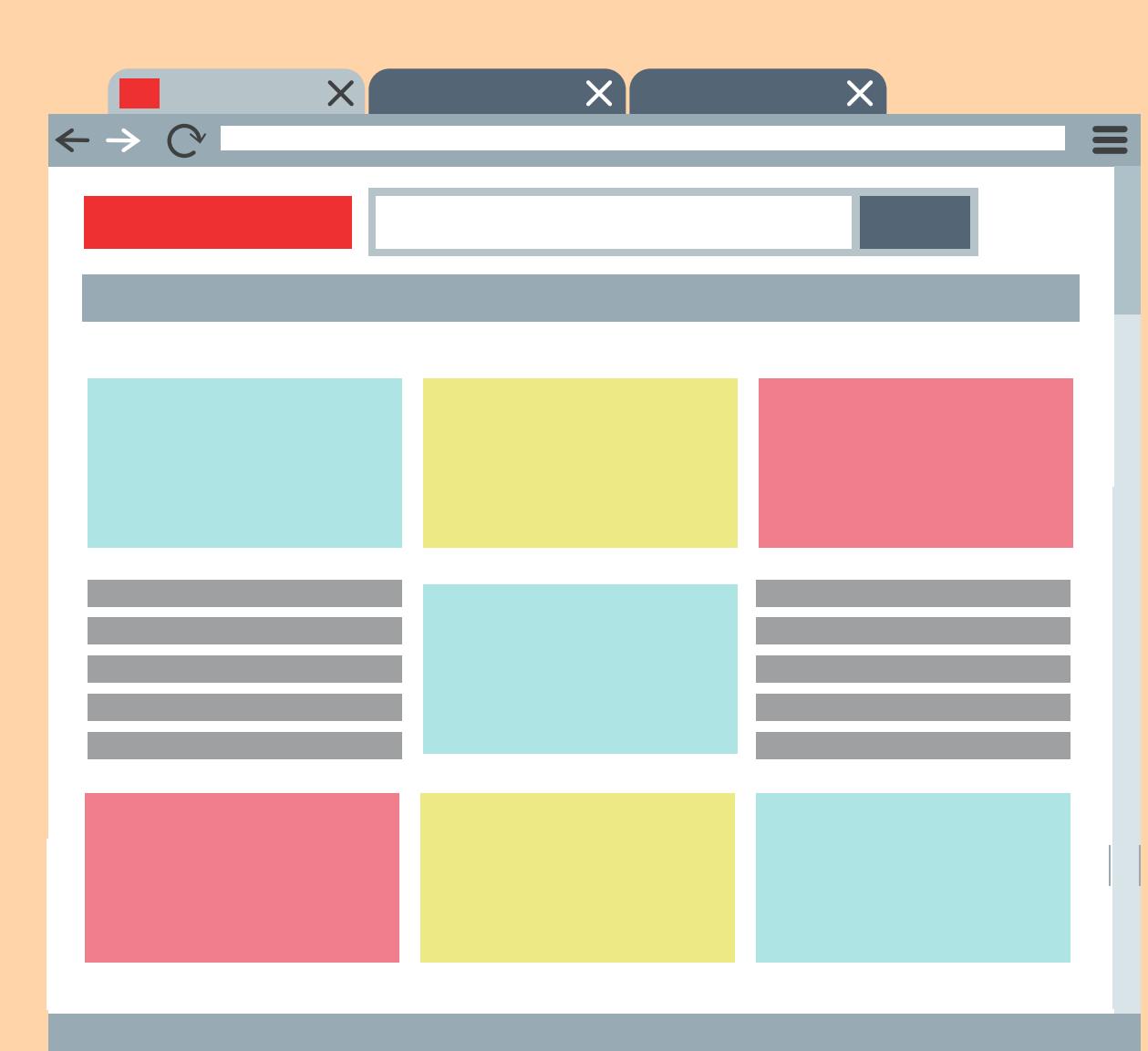
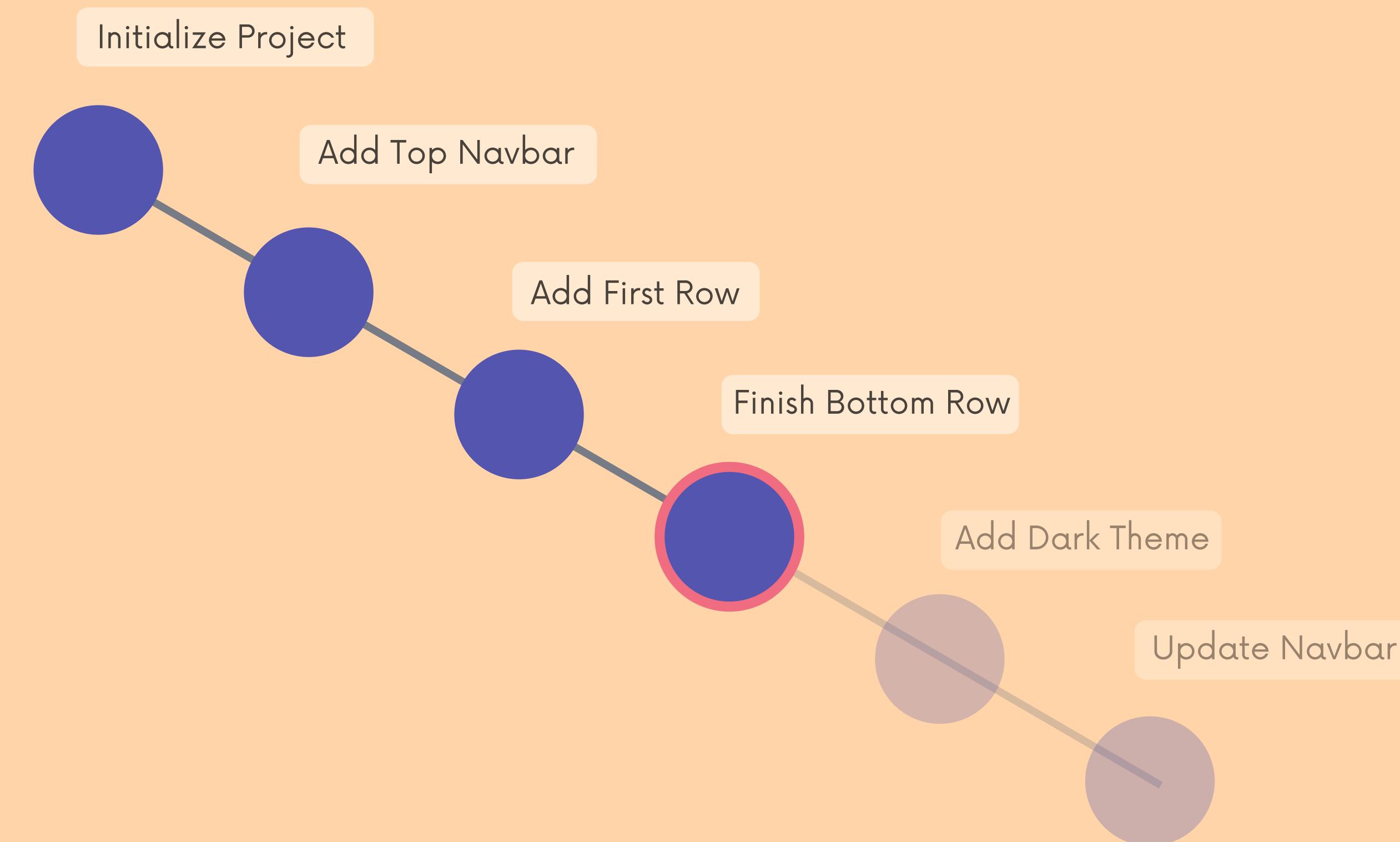
# No problem!



# I can go back to prior checkpoints I made!

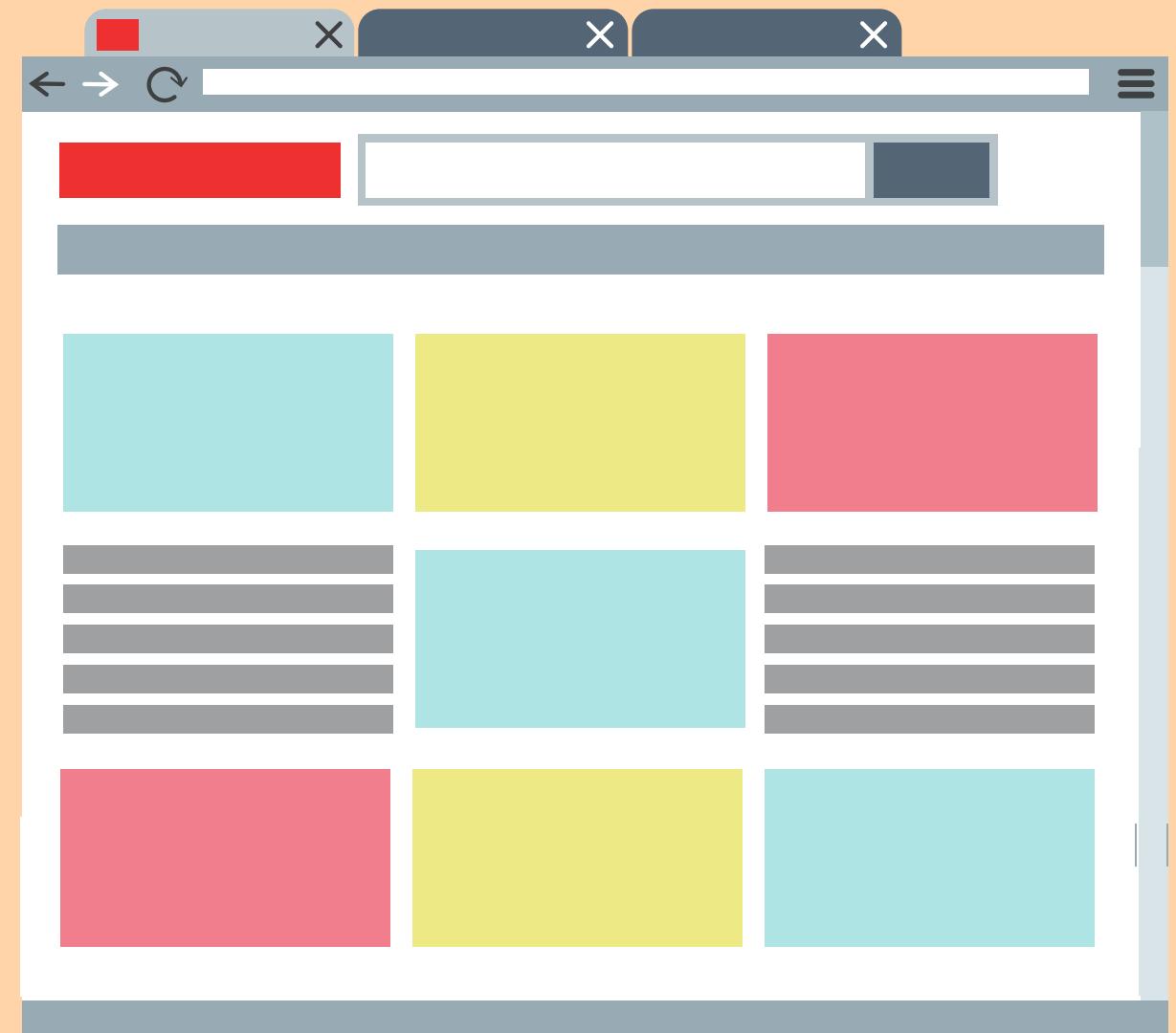
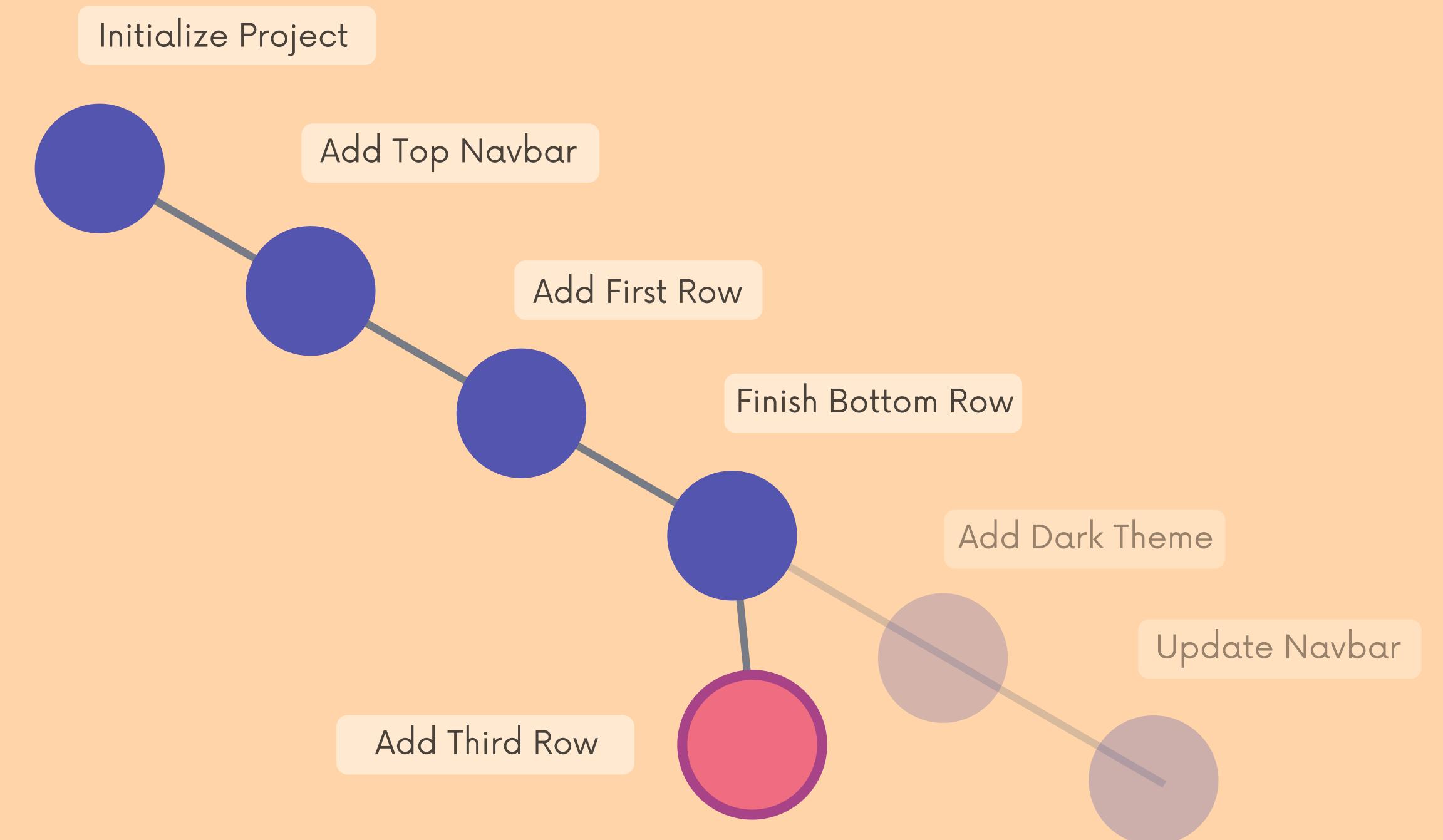


# I can even start more work off of an old checkpoint

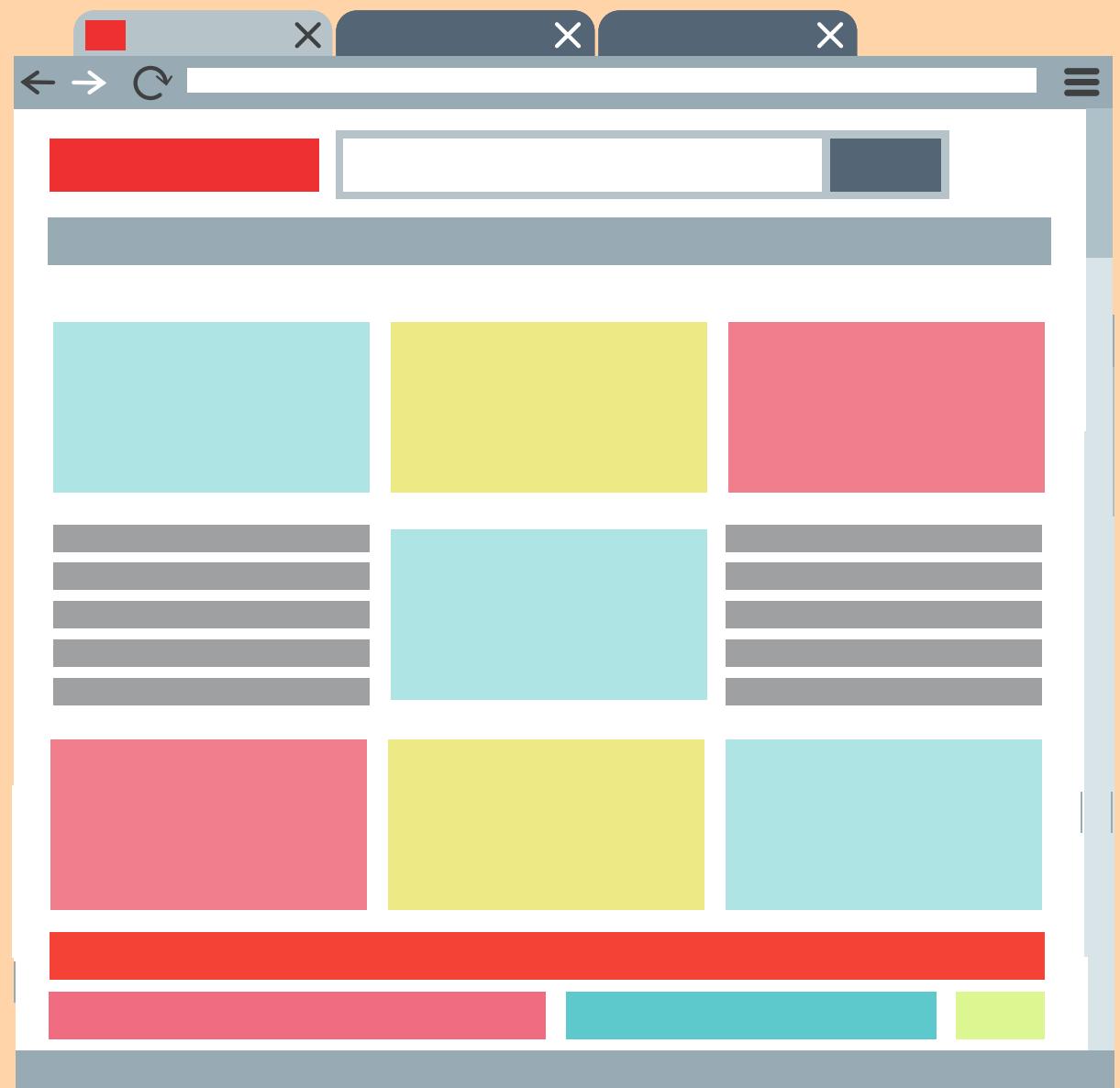
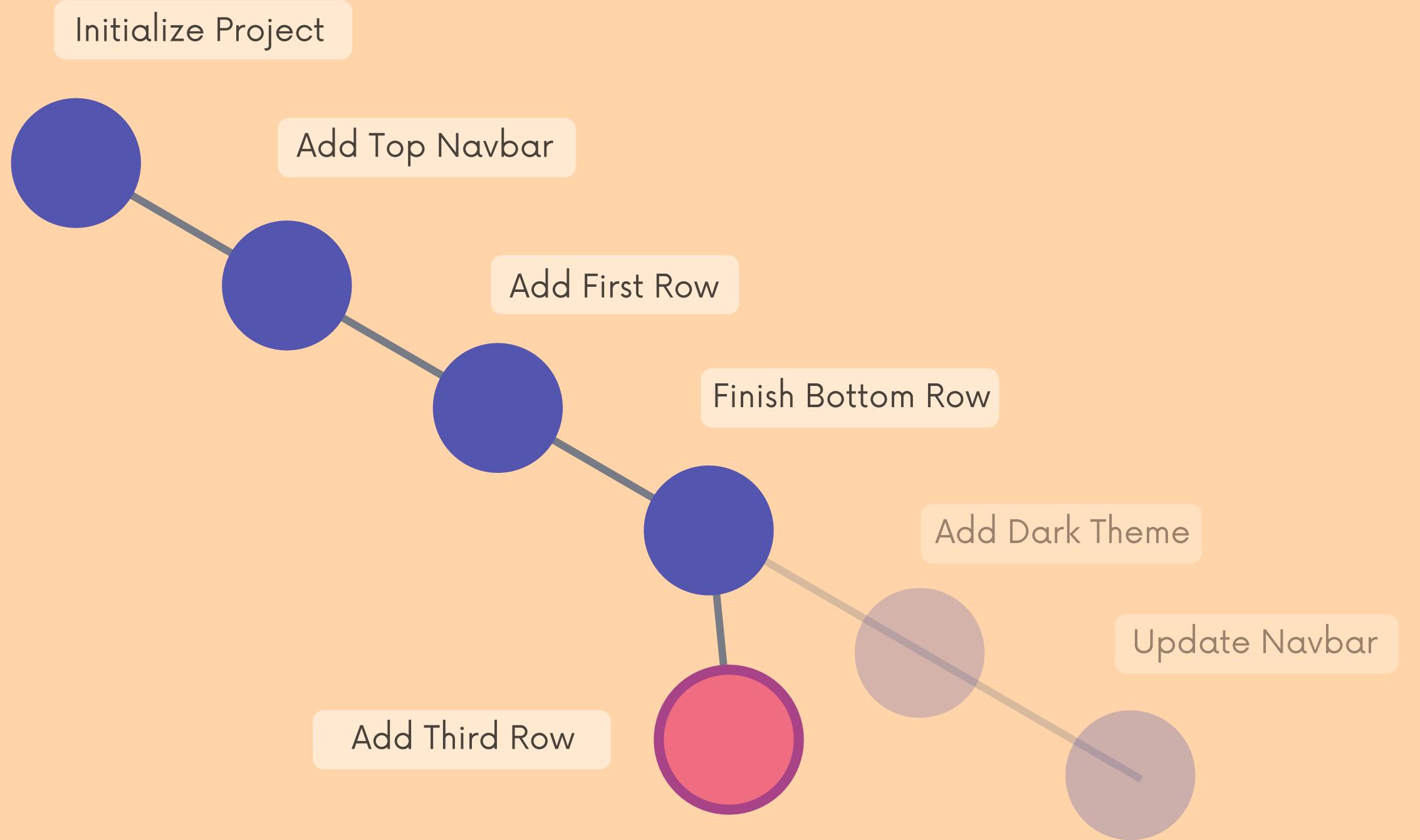


I add more content!

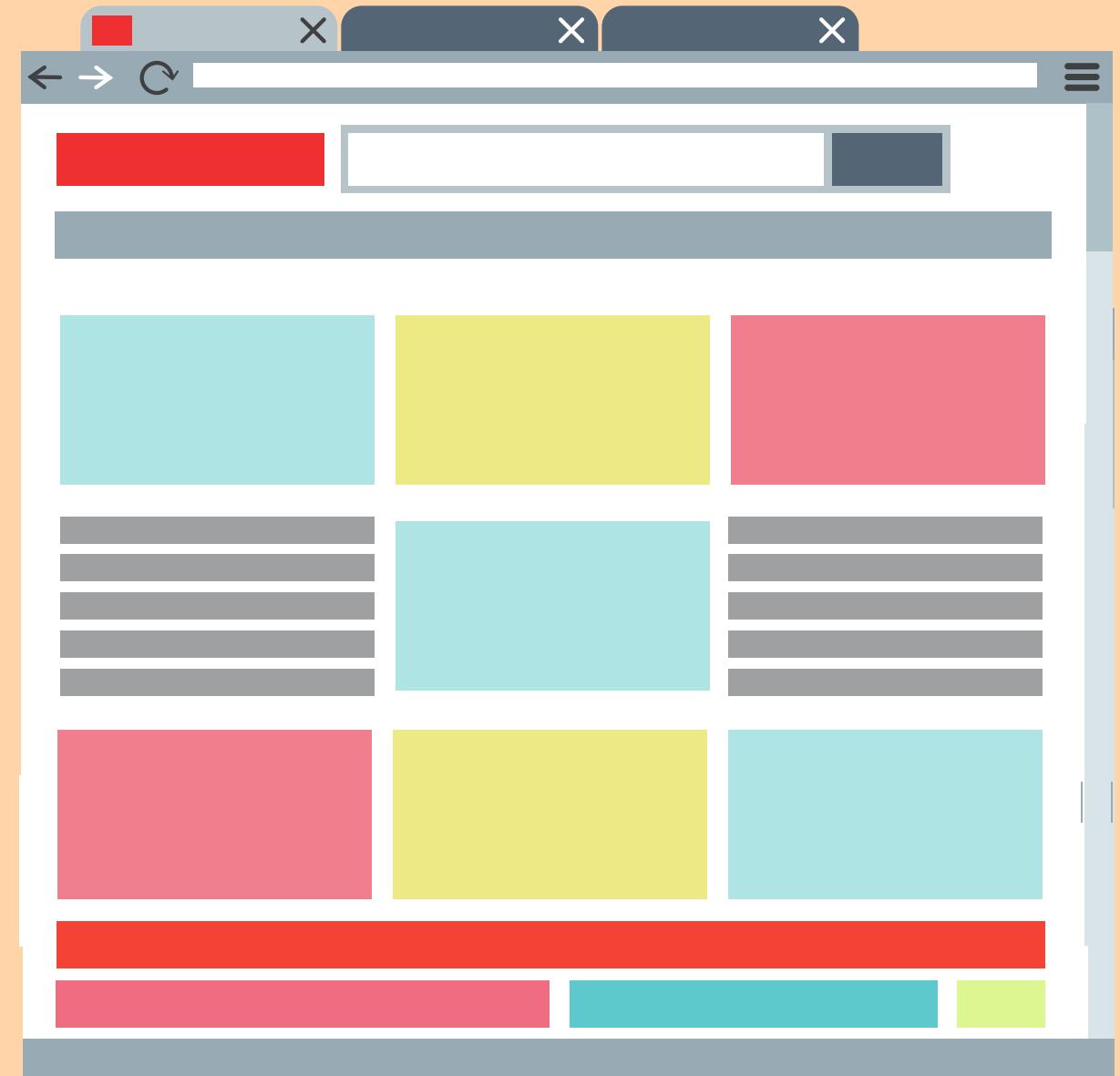
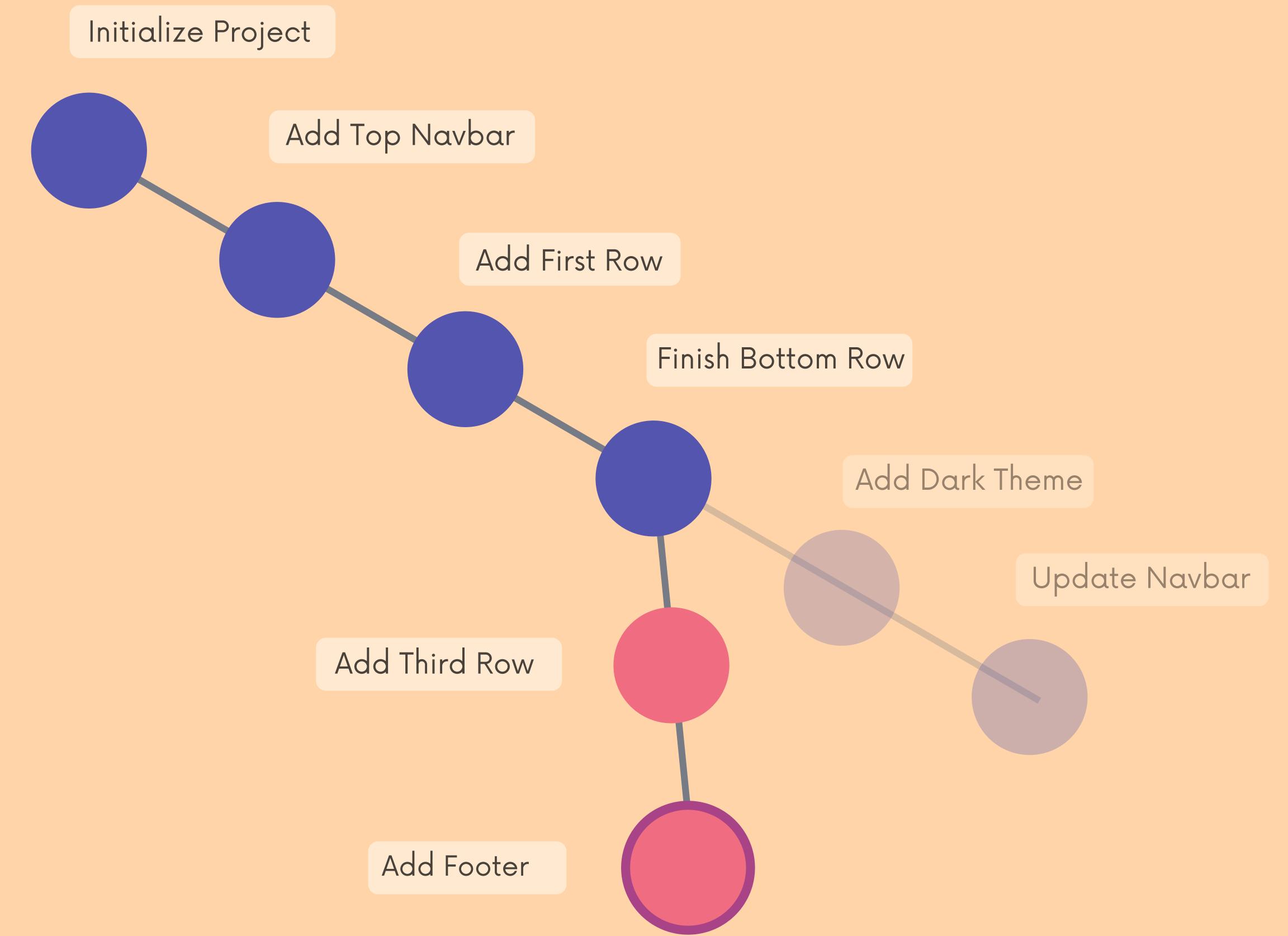
# I add a new checkpoint!



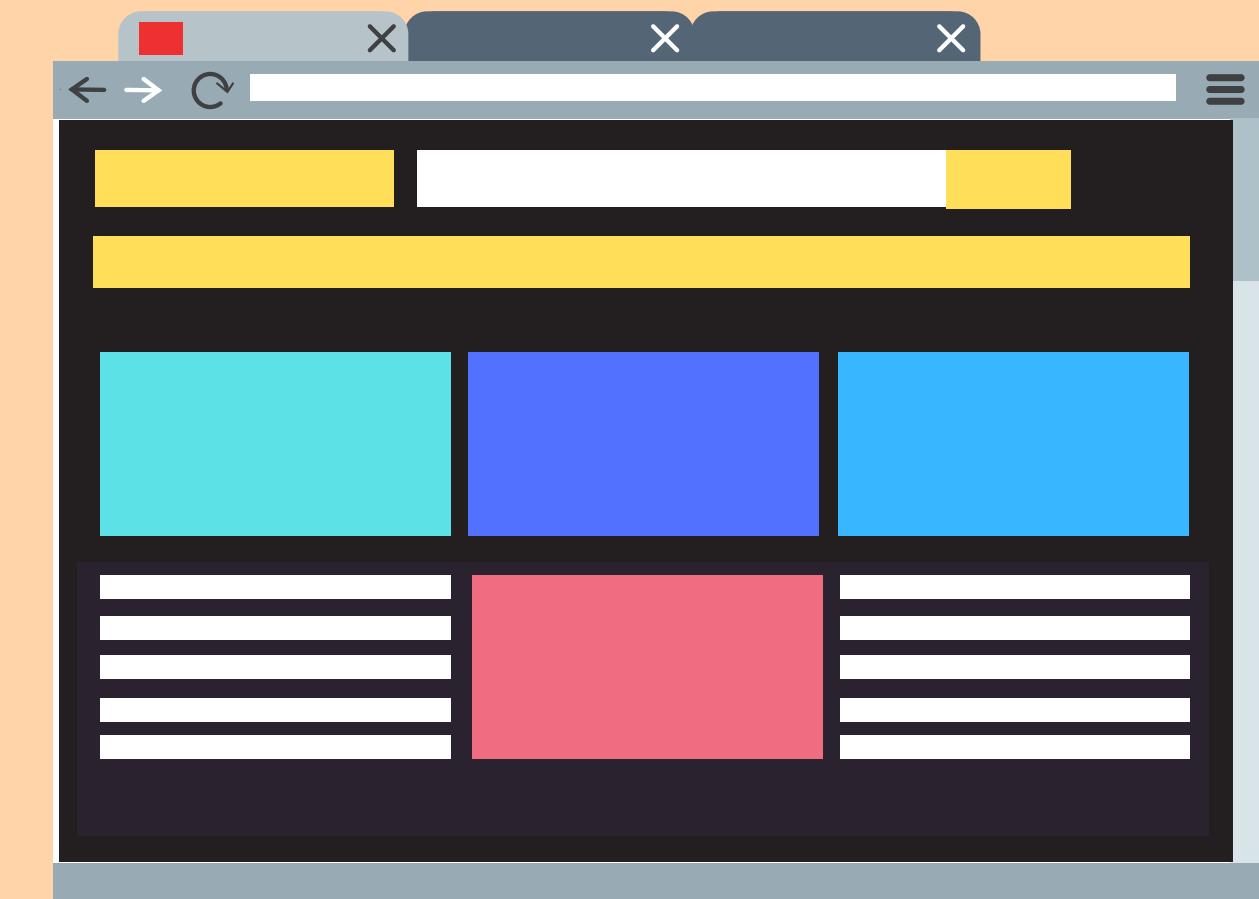
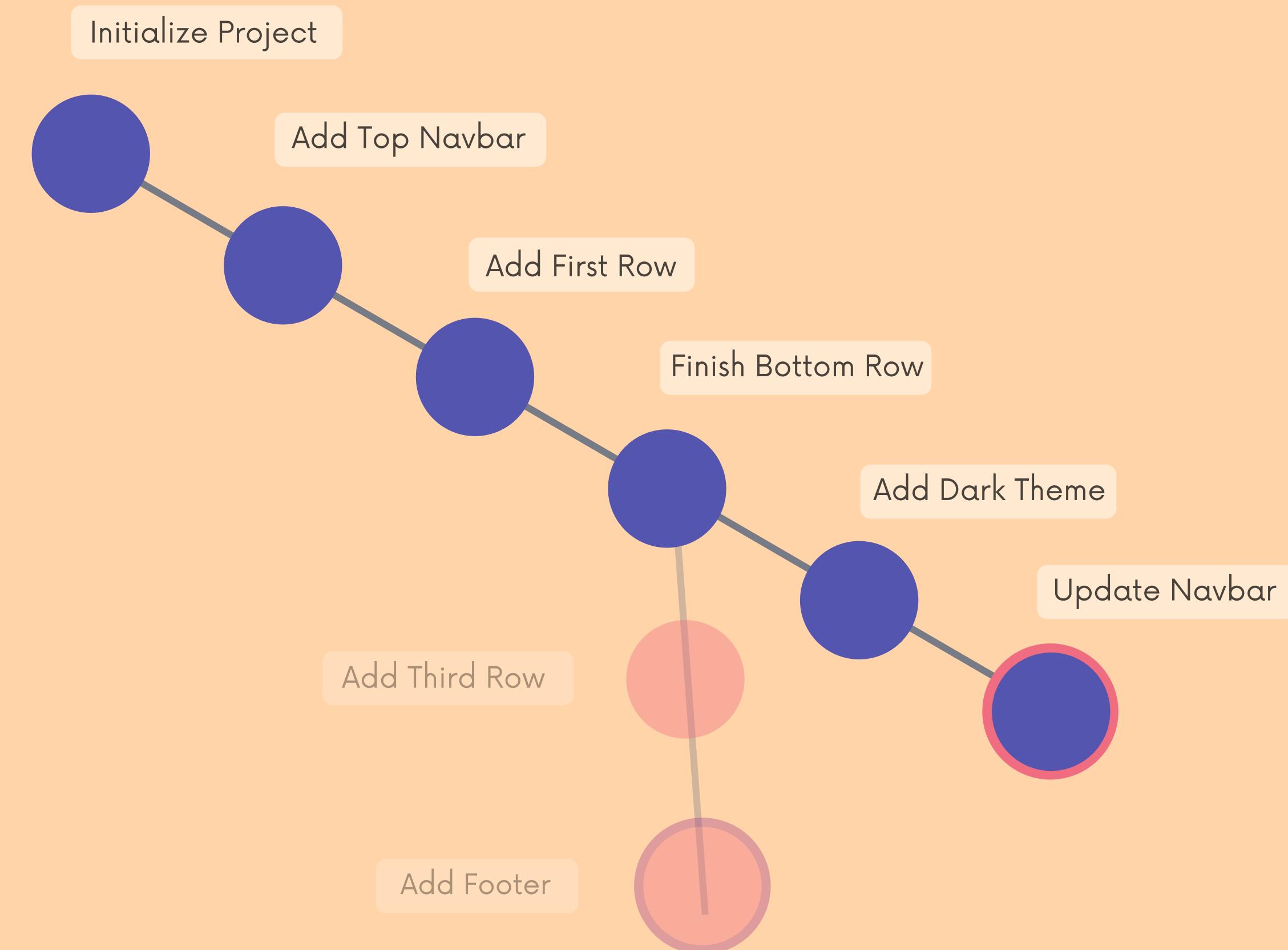
# I add a new footer



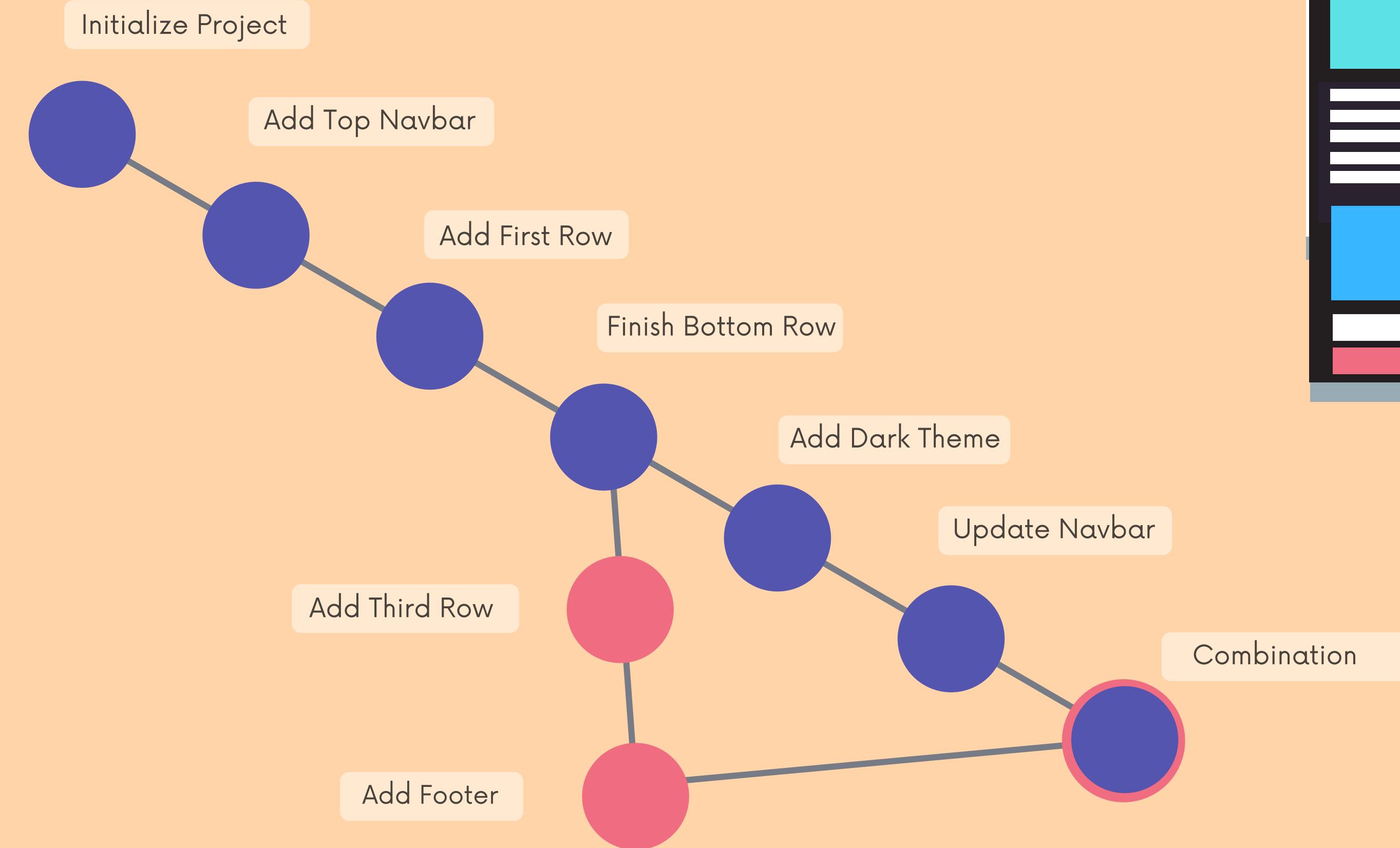
# Another checkpoint!



I can switch back to a  
dark mode checkpoint



# And I can even combine checkpoints!



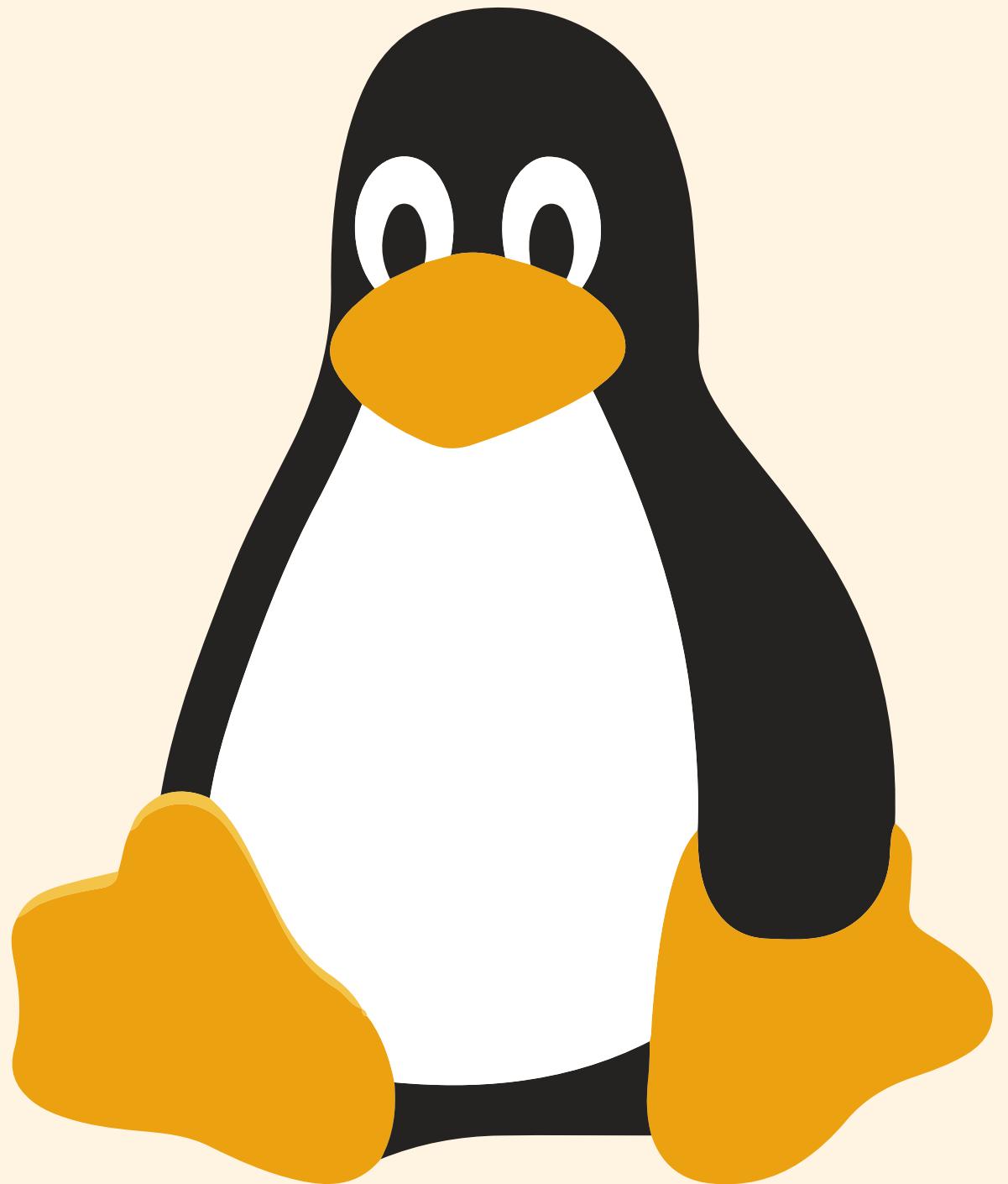


# A Tiny Bit of Git History

# Linus Torvalds

Linus Torvalds is a legendary software engineer. He is the creator and main developer behind Linux and Git!

In 2005, while working on Linux, he became frustrated with the available version control systems. The existing tools were slow, closed-source, and usually paid.





# The Birth Of Git

Torvalds wanted a version control system that was super fast AND free, unlike the existing tools.

On April 3rd 2005 he got to work on his own VCS, which would become Git. In a matter of days he had most basic functionality done.

The first official Git release came a couple months later. 15 years later in 2020, over 90% of developers worldwide use Git on a daily basis!



# Behind The Name

Torvalds referred to Git as "the stupid content tracker" while he was working on it. Eventually he settled on the name Git.

The official Git source code explains a couple different meanings for the name, depending on your mood:

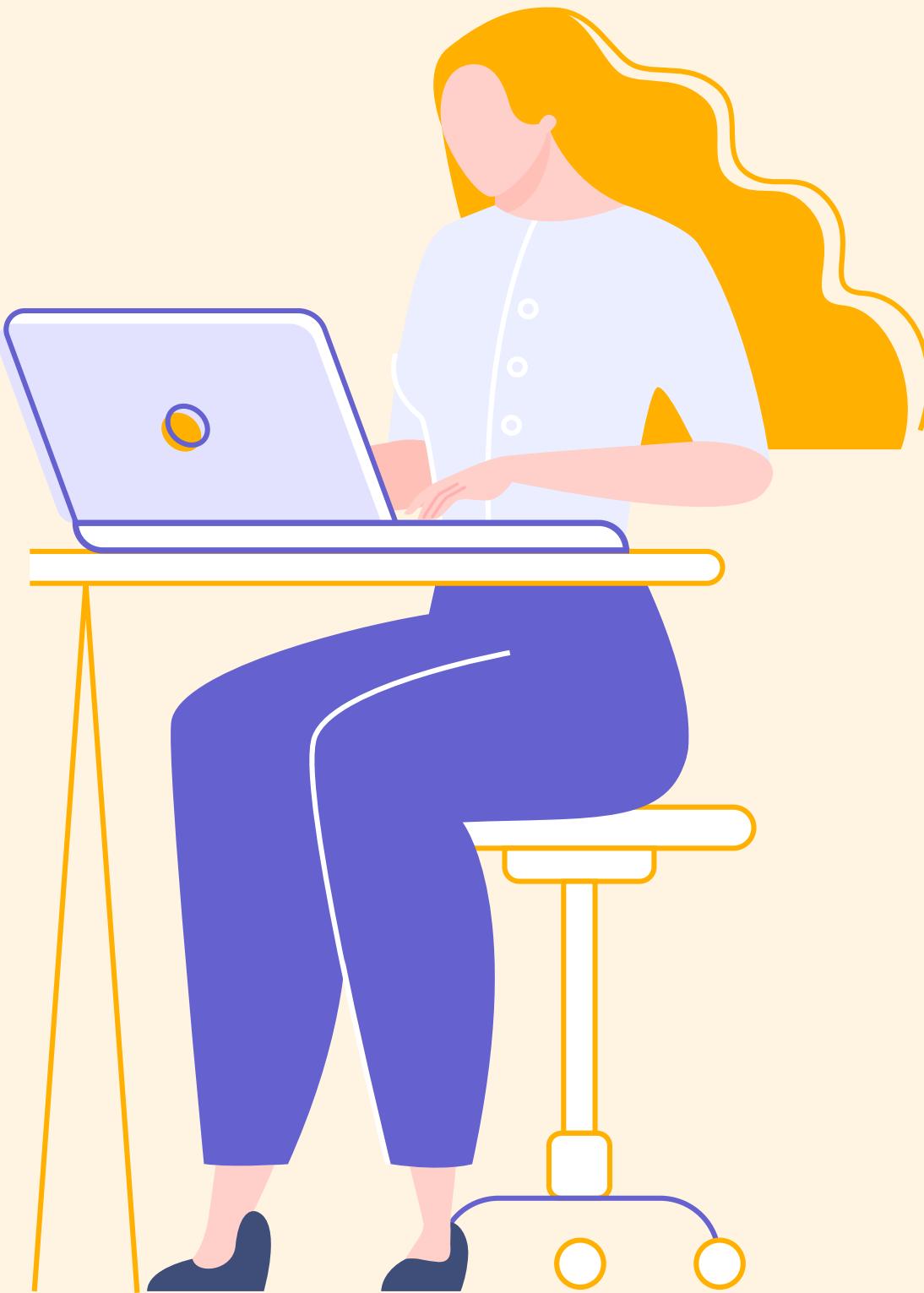
- a random three-letter combination that is pronounceable, and not actually used by any common UNIX command.
- stupid. contemptible and despicable. simple.
- "global information tracker": you're in a good mood, and it actually works for you. Angels sing, and a light suddenly fills the room.
- "g#ddamn idiotic truckload of sh\*t": when it breaks





# Engineers & Coders

From massive tech giants like Facebook and Google to the tiniest of startups, developers across the globe use Git. If you plan on becoming a developer, Git is essentially a must-have.





# Tech-Adjacent Roles

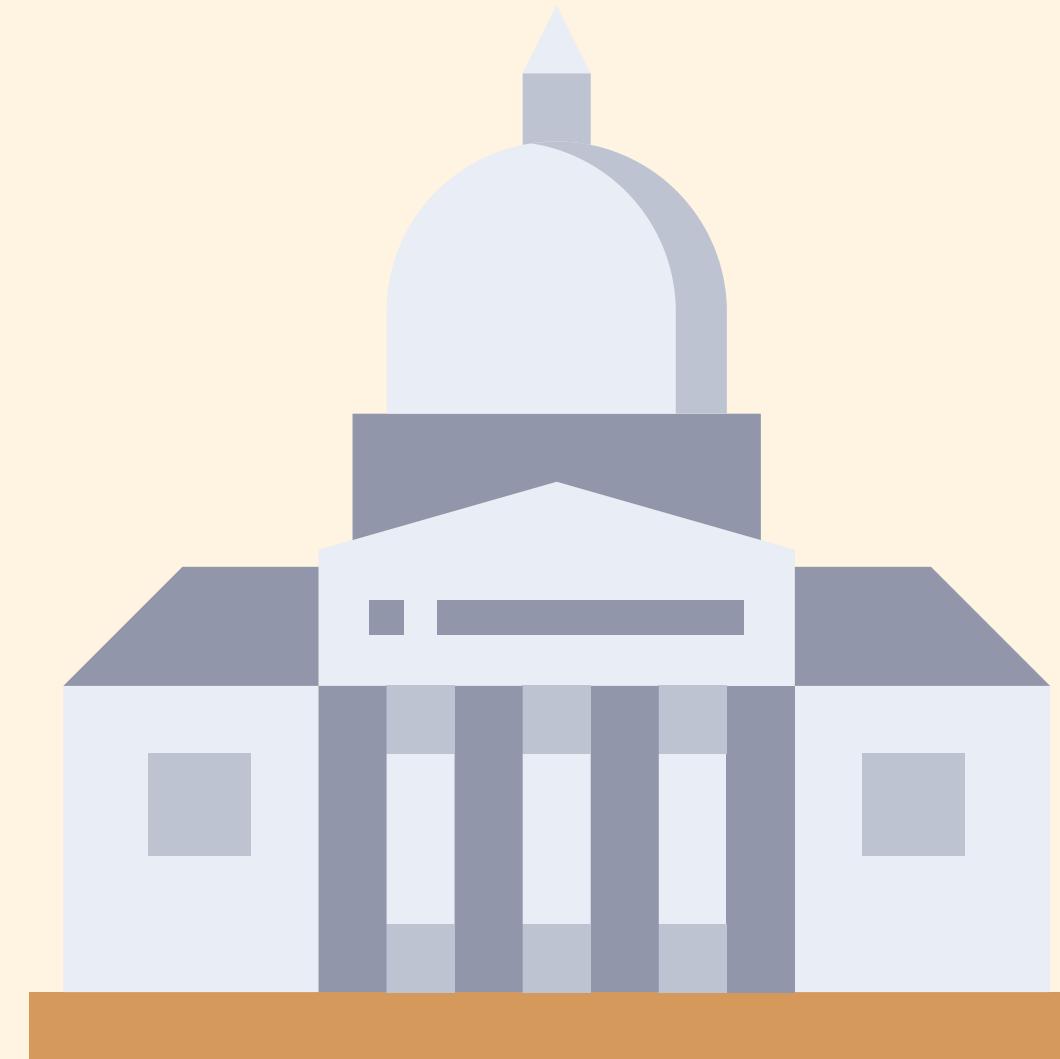
Many people in non-developer roles end up learning the basics of Git to collaborate with their coworkers. Designers in particular often need to work with Git.





# Governments

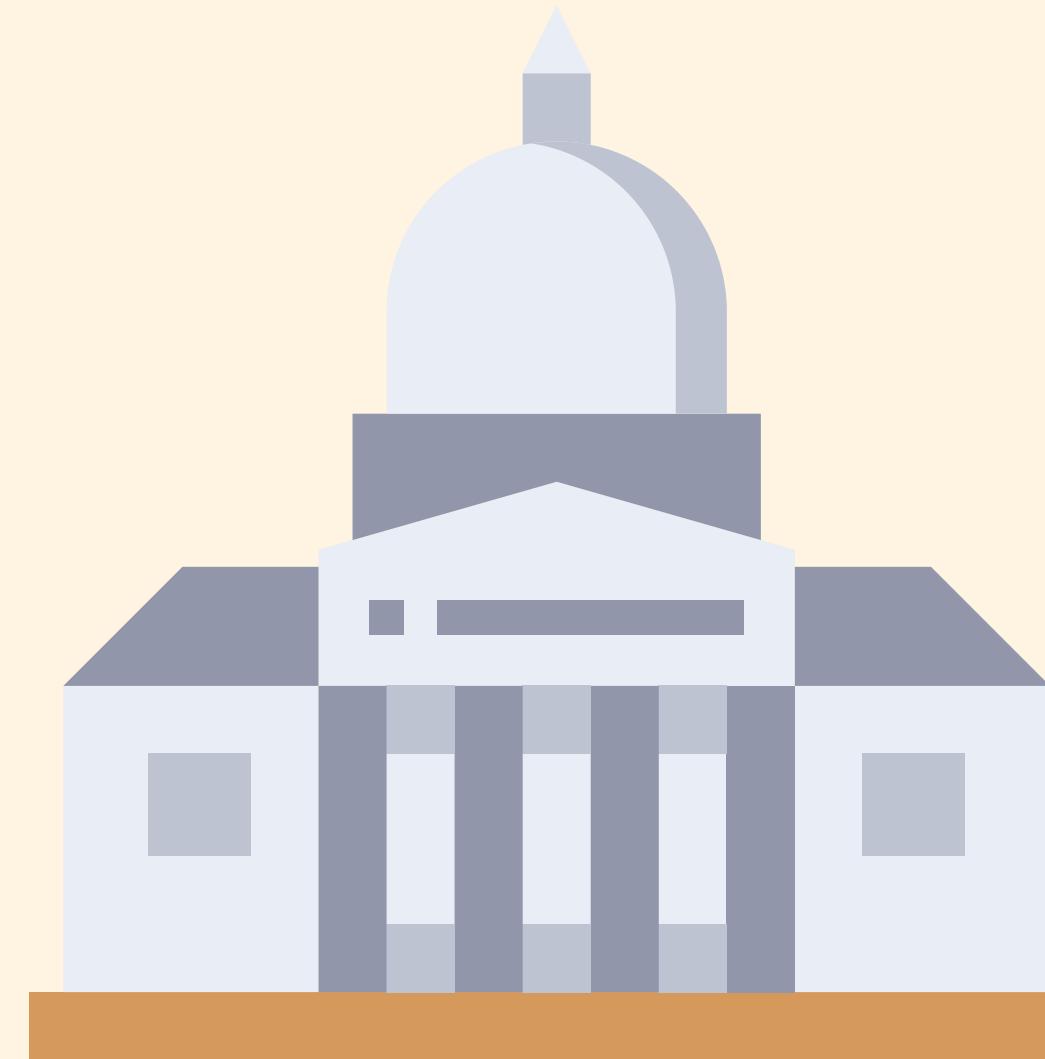
In recent years, governments have started using Git to manage the drafting of laws. Washington DC City council uses Git & Github to publish its laws. Citizens have even found and fixed typos collaboratively using Git!





# Scientists

Git is commonly used by research teams at universities and agencies around the world to manage code, data sets, and more. Git + Github is especially powerful in the peer-reviewed world of science and research.





# Writers

Some writers love using Git to manage drafts of complex novels, screenplays, or other works with lots of moving parts and constant changes across multiple files.

In particular, Git is gaining popularity for use in collaborative textbook writing with multiple authors.





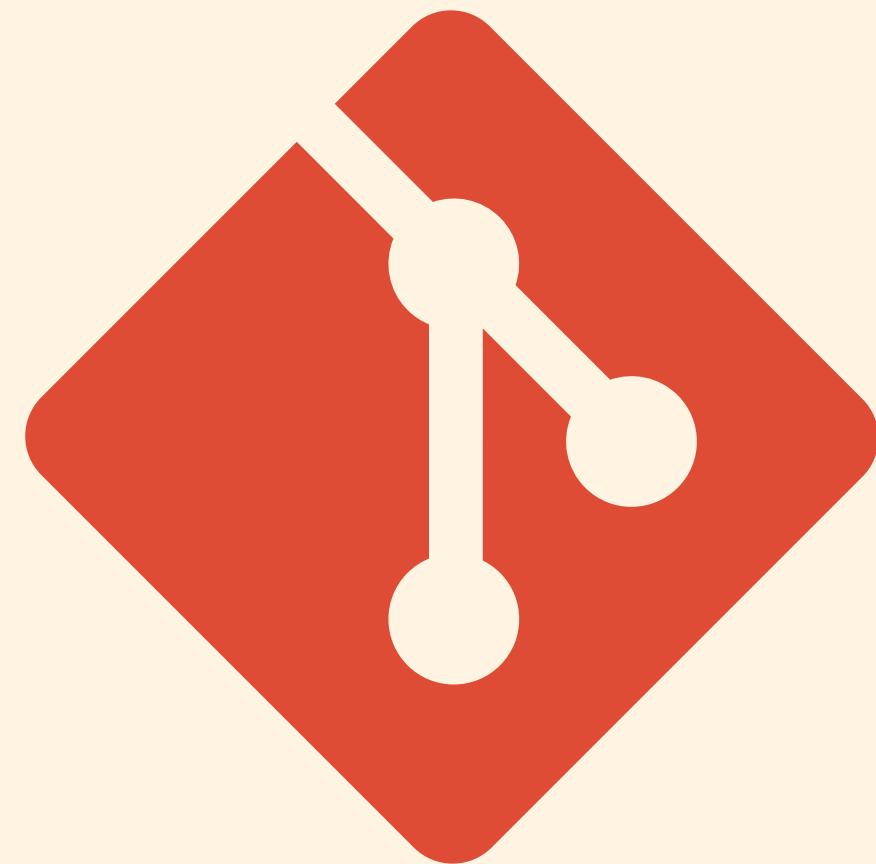
# Anyone Really

People find very creative uses for Git ranging from keeping a daily diary to drafting PhD theses to tracking changes to photoshop files. At least one composer writes his symphonies using Git!



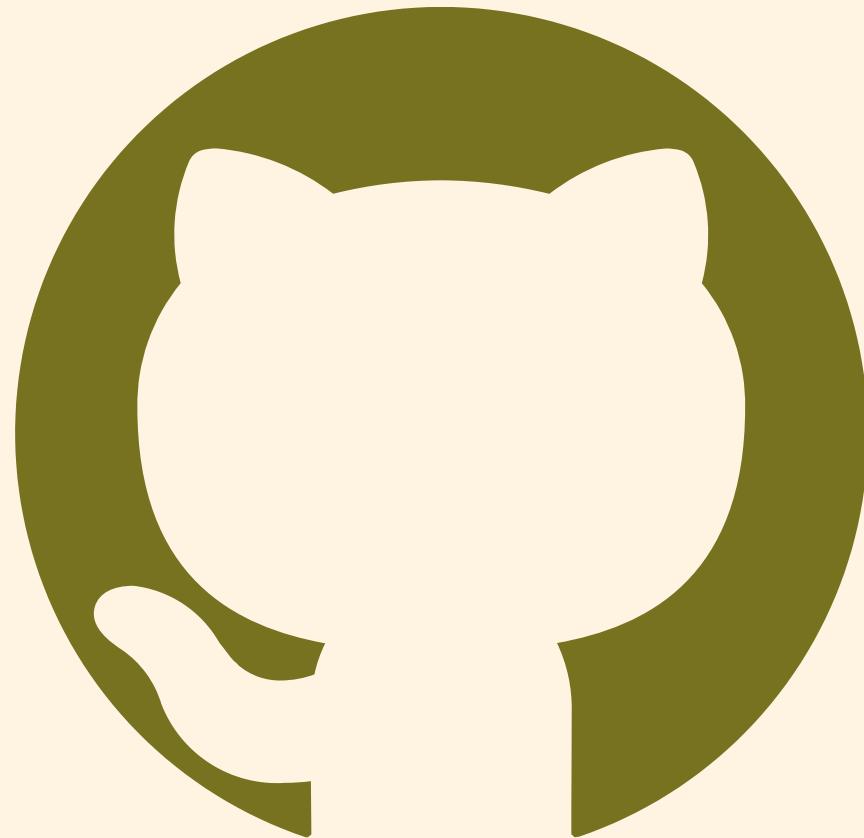
# Git ≠ Github





# Git

Git is the version control software that runs locally on your machine. You don't need to register for an account. You don't need the internet to use it. You can use Git without ever touching Github.



# Github

Github is a service that hosts Git repositories in the cloud and makes it easier to collaborate with other people. You do need to sign up for an account to use Github. It's an online place to share work that is done using Git.

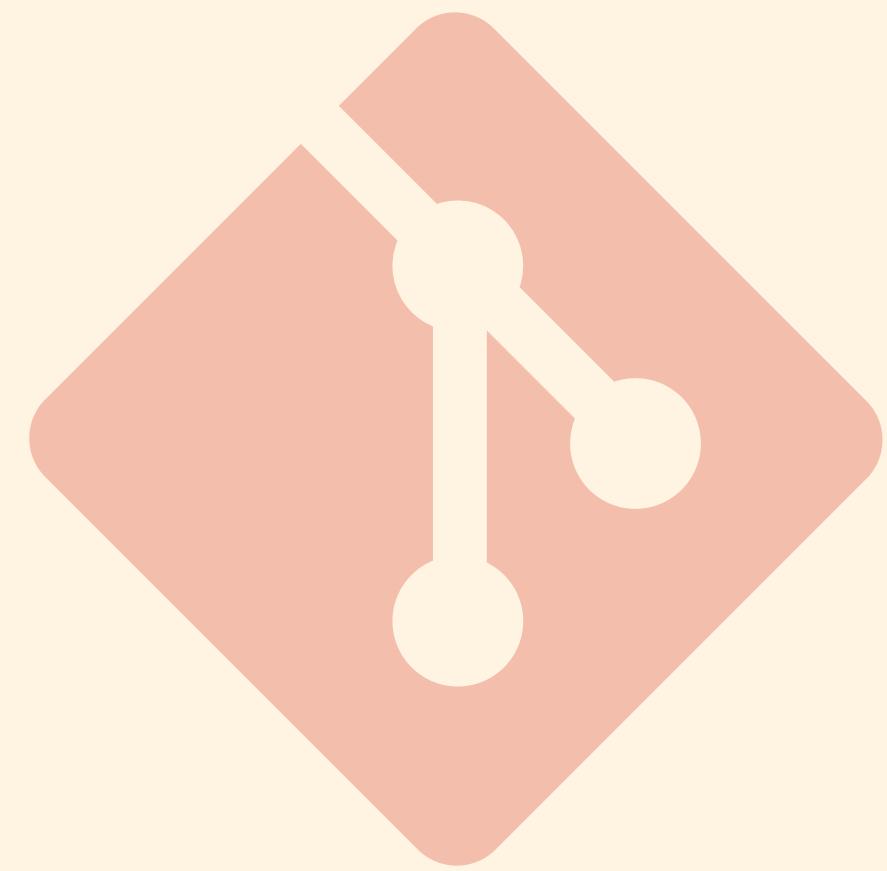


# Git

Git is the version control software that runs locally on your machine. You don't need to register for an account. You don't need the internet to use it. You can use Git without ever touching Github.

# Github

Github is a service that hosts Git repositories in the cloud and makes it easier to collaborate with other people. You do need to sign up for an account to use Github. It's an online place to share work that is done using Git.



# Git

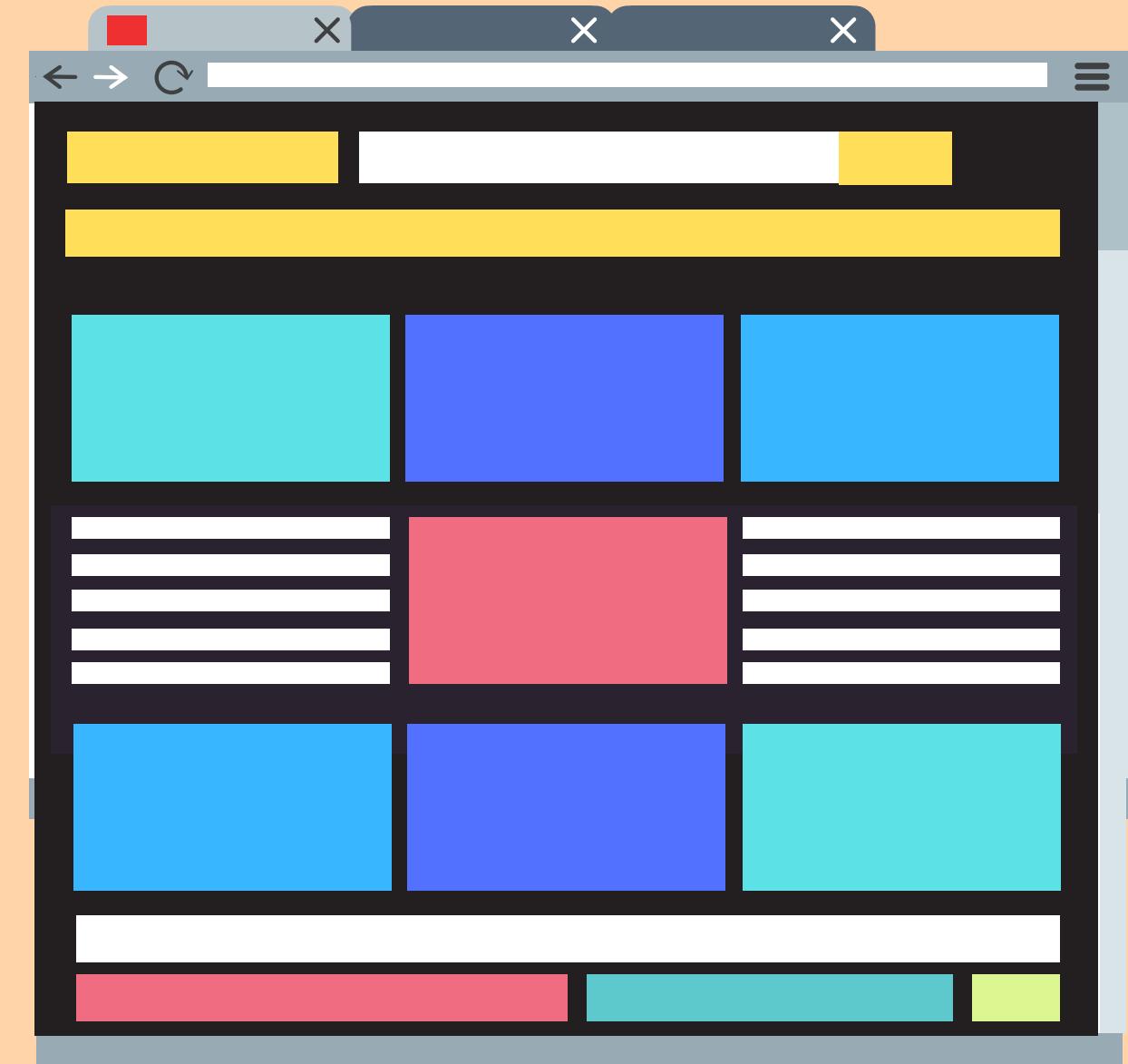
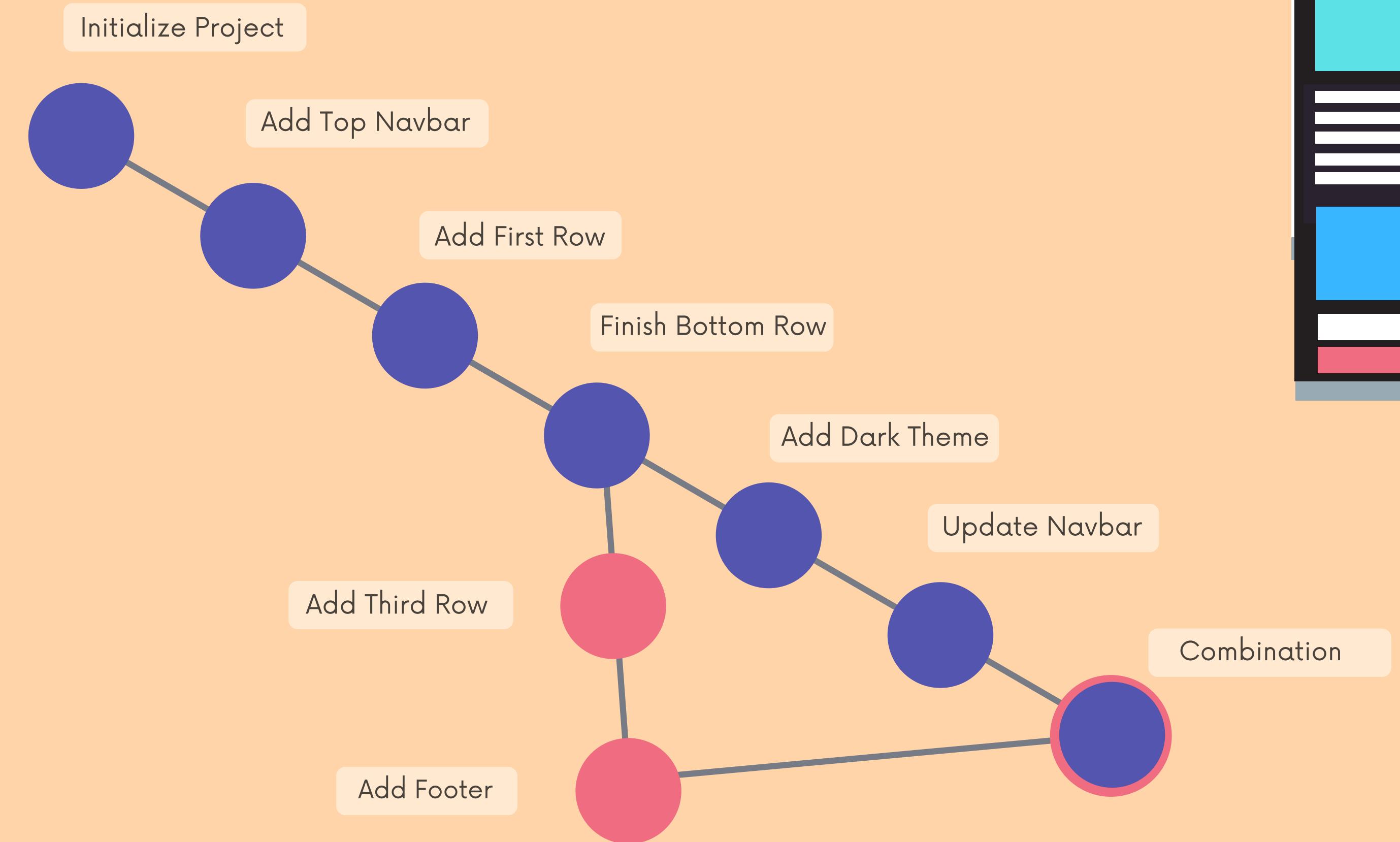
Git is the version control software that runs locally on your machine. You don't need to register for an account. You don't need the internet to use it. You can use Git without ever touching Github.



# Github

Github is a service that hosts Git repositories in the cloud and makes it easier to collaborate with other people. You do need to sign up for an account to use Github. It's an online place to share work that is done using Git.

This is all on my local machine



# I can upload my history to Github to share with others

