Full-Stack: Final project

Full-Stack – Final Project

The following exercise contains the following subjects:

- EVERYTHING

Submitting instructions:

Please add the following to hive:

- A link to the repository
- Free text – a description of the app. Stuff you found hard to implement, known bugs and your review of this assignment.

You must use Git throughout the assignment (and not only commit at the end)

Use all the tech we have learned

- HTML/CSS
- JS
- React
- Node js
- MongoDB

Planning phase

Pain / Central focus

It is the center focus that everything in your project is built for.

Every good piece of software has a way to fix a pain point in people's lives. Netflix removed the 'pain' of traditional movie rentals, Facebook allowed its users to feel connected from anywhere in the world.

However, If you just want to build something that interests you, that's *amazing, and* don't let something like finding 'pain points' stop you. However, every piece of software should have a central focus. so find that before you start building anything!

Solution / features

Keeping our pain / central focus point in mind, we can start to dive into what cool features we want our software to be capable of doing to solve the pain / central focus. As we create this primary 'layer' of features, we should constantly be asking ourselves, "Is this feature going to help solve our pain point/ central focus?" If not, it's probably a waste of our time to code it.

Time management

Have 3 categories of features you want to implement.

1. Must do features.
2. Features you want to implement if you have the time.
3. Future features that would be nice to have.

Start with the back:

Database Modeling

Setup your database. What are the collections and records do you need?

API Endpoint planning

Plan your end points. Do you need full CRUD endpoints to each model? Are you going to handle query parameters with your API? If you have authentication, do some of the endpoints need to be protected?

Test all your end points on with Postman.

Then go to the Front

Design:

Components

Think of the components you need for your features.

Where to place your state.

What utility functions do you need to achieve your features.

Routes

Think what routes would you need for your components.

Pages Layout

Sketch your pages interfaces and think about components should belong where.

Keep it DRY

When you look over your blueprint, you should start to see some patterns in reused functionality. This is our chance to build out the reusable bits of code your project will be using. By doing this we are using DRY (Don’t Repeat Yourself!) coding practices that will not only speed up our development process but make it more maintainable and improve our code quality as well! *It is arguably the most important step*. Think of this portion as a tiny code library specifically for your project filled with methods, functions and components that will make your lives easier.

Push To Heroku

After each feature, push it to Heroku so you know it works in production as well.

Ideas and inspiration:

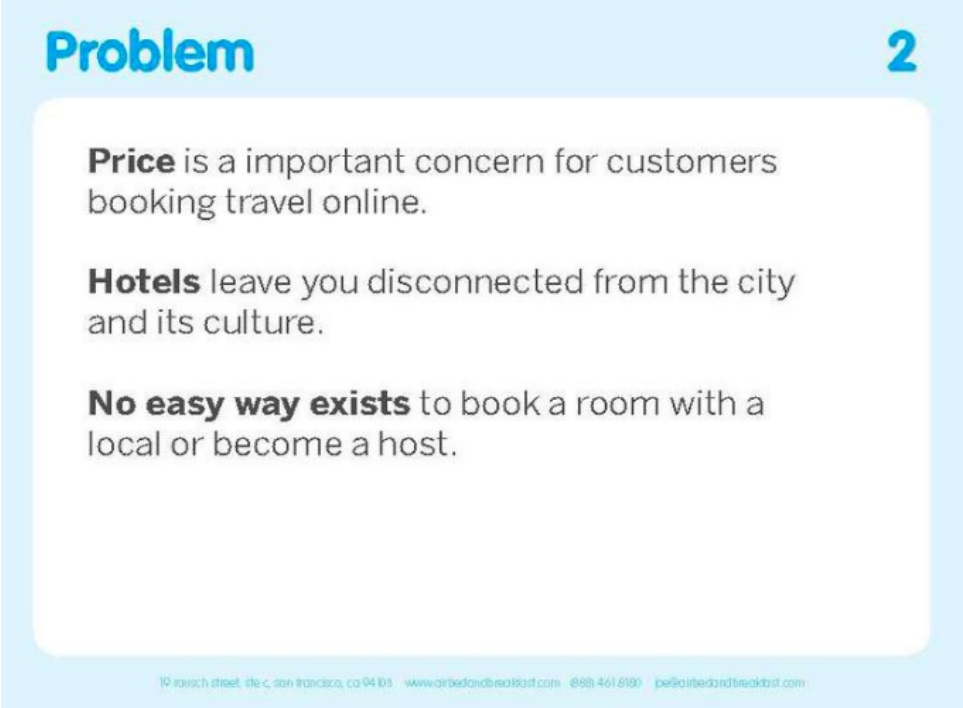
https://holycoders.com/full-stack-web-development-coding-project-ideas/

https://dev.to/dailydevtips1/5-full-stack-projects-to-add-to-your-portfolio-before-2020-ends-42h3

Airbnb’s Pitch

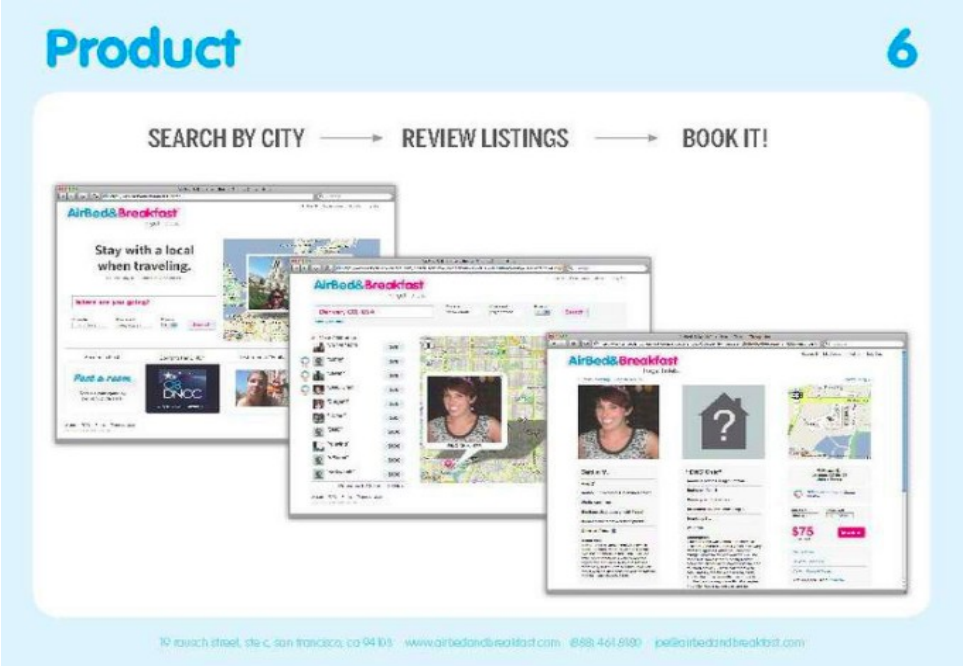
Lets look at a few slides that is relevant for us on how AirBnb did their famous pitch in 2009 that raised them $600k.

Problem/ Pain



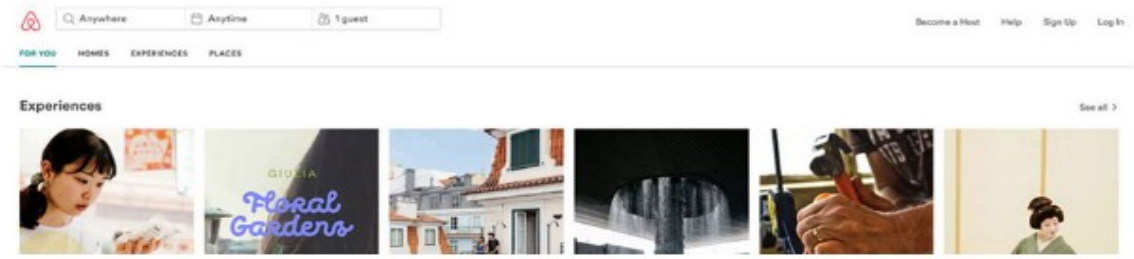
Solution

Product / core features



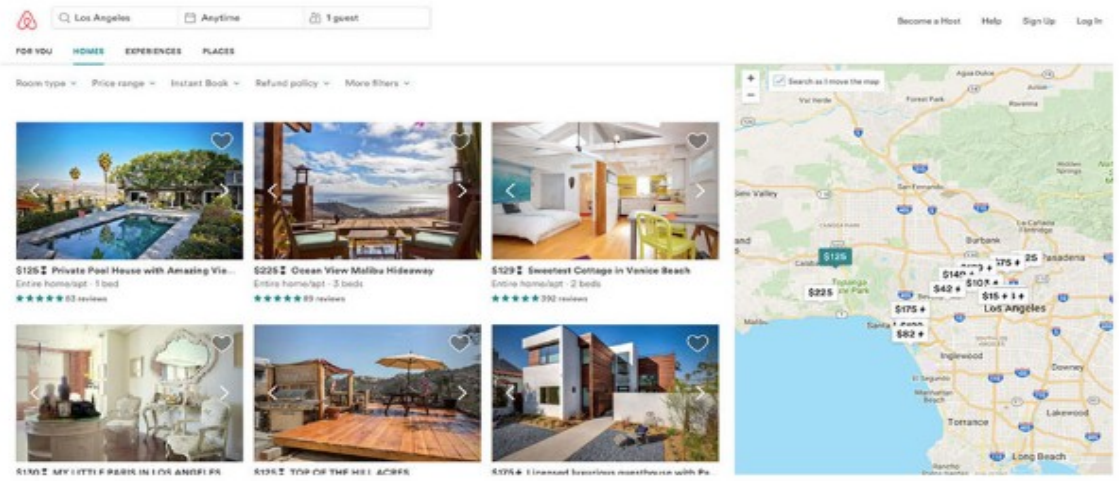
airbnb

Search by city



airbnb

Review listings



airbnb

Book it!