

LECTURE NOTE 5 [CSC 421]

BUILDING A WEB APPLICATION (APP)

5.1 Prerequisites for Building a Web Application

To make a data-centric web app from the bottom-up, it is advantageous to understand:

- Backend language (e.g., Python, Ruby) - control how your web app works
- Web front end (HTML, CSS, Javascript) - for the look and feel of your web app
- DevOps (Github, Jenkins) - Deploying / hosting your web app

If you do not have any experience with the points above, you need not worry.

You have two options:

1. Learn the points above - there are lots of resources online to help you. Codecademy and Udemy are recommend.
2. Use a web app builder like Budibase - As a builder, Budibase will remove the need to learn a backend language. On top of that, Budibase will also take care of a lot of your DevOps tasks such as hosting.

5.2 Building a Web Application

5.2.1 Step 1 – Source an idea

Before making a web app, you must first understand what you intend to build and more importantly, the reason for it.

Your idea should stem from solving someone's problem. Ideally, your own problem.

It is important that developer choose an idea which interests him /her. Ask yourself:

- How much time do I have to build this app?
- What am I interested in?
- What apps do I enjoy using?
- What do I like about these apps?
- How much time/money will this app save or generate for me (as a user)?
- How much will it improve my life

5.2.2 Step 2 – Market Research

Once you have chosen your idea(s), it is important to research the market to see:

1. If a similar product exists
2. If a market exists

The number 1 reason start-ups fail, is the failure to achieve product market fit. *“Product/market fit means being in a good market with a product that can satisfy that market.”*

To quickly find out if a similar web app exists, use the following tools to search for your idea:

- Google
- Patent and trademark search
- Betalist
- Product hunt

If a similar product exists, do not worry. This can be a sign a market for your new idea exists. Your future competitors have laid the groundwork, educated the market. It is time for you to swoop in and steal the thunder. If a similar product does not exist, it is a possibility you have struck lucky. On the other hand, it is a possibility someone before has ventured down this path and hit a dead-end.

Nobody wants to experience that, so it is important to dive deep into the market and source the wisdom of:

1. Your Web App's target market: Share your web app idea on forums related to your target market. If you know anyone who works within your target market, explain your idea to them. The more you talk and receive validation from your target market, the better.
2. Google Trends: A quick search of your web app idea will reveal relating trends.
3. SEO tool: MOZ/Ahrefs is recommended. Google's keyword planner will suffice. Write a list of keywords relating to your web app. If it is an 'OKR tool', use the tools to search 'OKR tool', 'OKR app', and 'objectives and key results software'. If the SEO tool indicates there are lots of people searching for your keyword terms, this is a small indicator you have a target market.
4. Social Media: Jump over to Twitter/Facebook groups and present your idea to your target market.
5. Events: If there is a local event in your area attracting people from your target market, go to it. Share your idea and record the feedback.

After completing the above steps, you should have enough information to understand if there is a market for your product. If there is a market for your product, and there is also established competition, it is important to research them.

5.2.3 Step 3 – Define your web apps functionality

You have got your idea, you have validated the market, it is now time to list everything you want your app to do.

A common mistake here is to get carried away. The more functionality you add, the longer it will take to build your web app. Quite often, the longer a web app takes to build, the more frustration you will experience.

Only define functionality which solves your target markets problems. Remember, your web app is a work in progress and the first goal is version 1. It will still have cool features and delight your users, but you must keep things simple.

For direction, I have included a list of basic functions required for a simple CRM app.

- ❖ Users can create an account
- ❖ Users can retrieve lost passwords
- ❖ Users can change their passwords
- ❖ Users can create new contacts
- ❖ Users can upload new contacts
- ❖ Users can assign a value to contacts
- ❖ Users can write notes under contacts
- ❖ Users can label a contact as a lead, customer, or associate
- ❖ Users can filter contacts by lead, customer, or associate
- ❖ Users can view the total value of leads, customers and associates

The above list will help you define your features. Once you are done, roll up your sleeves. It is time to get creative! Moving from the Ideation stage, to design stage.

5.2.4 Step 4 - Sketch your web app

There are multiple stages of designing a web app. The first stage is sketching using a notebook (with no lines) and pen/pencil. After step 1, 2 and 3, you should have an idea of what your web app is, who your users are, and the features it will have.

Sketch out the wireframe of your web apps UI (User Interface) - it does not have to be exact - this is just a sketch.

When sketching, consider the following:

- Navigation
- Branding
- Forms
- Buttons
- Any other interactive elements

Sketch different versions of your web app. Consider how your web app's functionality will affect the overall design. Annotate your sketch and outline how your app should work.

Taking notes will help you clarify and understand why you have designed certain elements at a later stage. Overcomplicating the design at this stage will only lead to frustration.

5.2.5 Step 5 - Plan your web apps workflow

It is time to put yourself in the shoes of your user. Here, we are going to plan your web apps workflow. Now is the time to go back to step 2 and look at your market research. Take your list of competitors and sign up to their free trials.

Have a quick play around with their product. Write notes on what you thought was good and what you thought was bad. Pay particular attention to the workflow.

After you have finished analyzing your competitor's web apps, it is time to write down different workflows for your app. Consider the following points:

- ✓ How does a user signup?
- ✓ Do they receive a verification email?
- ✓ How does a user log in?
- ✓ How does a user change their password?
- ✓ How does a user navigate through the app?
- ✓ How does a user change their user settings?
- ✓ How does a user pay for the app?
- ✓ How does a user cancel their subscription?

All of a sudden, our one-page web app turns into a 10-page web app. Write a list of all the different pages your web application will have. Consider the different states of pages. For example, the homepage will have two states; logged in and logged out. Logged in users will see a different page than logged out users.

5.2.6 Step 6 – Wireframing / Prototyping Your Web Application

Ok, it is time to turn those sketches and that new-found understanding of your web application into a wireframe/prototype.

Wireframing is the process of designing a blueprint of your web application while Prototyping is taking wireframing a step further, adding an interactive display.

The decision to wireframe or prototype is up to you. If you have the time, I would have recommended prototyping as it will make it easier to communicate your web app when seeking validation. You can prototype/wireframe using the following tools:

- Sketch (macOS)
- InVision Studio (macOs)
- Adobe XD (macOS, Windows)
- Figma (Web, macOS, Windows, Linux)
- Balsamiq (macOS, Windows, Web)

I recommend you create a design system/style guide first. You can find inspiration at UXPin. Design systems improve design consistency. But it's not required.