

IDEA and USERS

What problem(s) does your app solve?

Removes the sharp learning curve required for complicated inventory trackers.

Improve user setup speed dramatically.

How does it solve those problems?

The application will use a simple interface without any complex features

Who is your target user?

Small business owners. Those that don't need advanced features and need a fast solution.

How much experience do they have with technology?

Minimal.

FEATURES

Write use cases for your app.

- For example, as a user I can store a list of my favorite food.

Use color to separate features by 'Minimum Viable Product' and 'Bonus features'

As a user I can...

...enter a name, price, and image URL for a new product in a form.

...click the 'Add to Inventory' button to add the new product to my inventory and clear the form.

...click the 'Cancel' button to clear the form without adding the product.

...view all the products in my inventory

As a user I can...

...click the 'Delete' button to delete a product.

...click the 'Edit' button to edit the product in the form.

...click the 'Save Changes' button to save the changes made to the product.

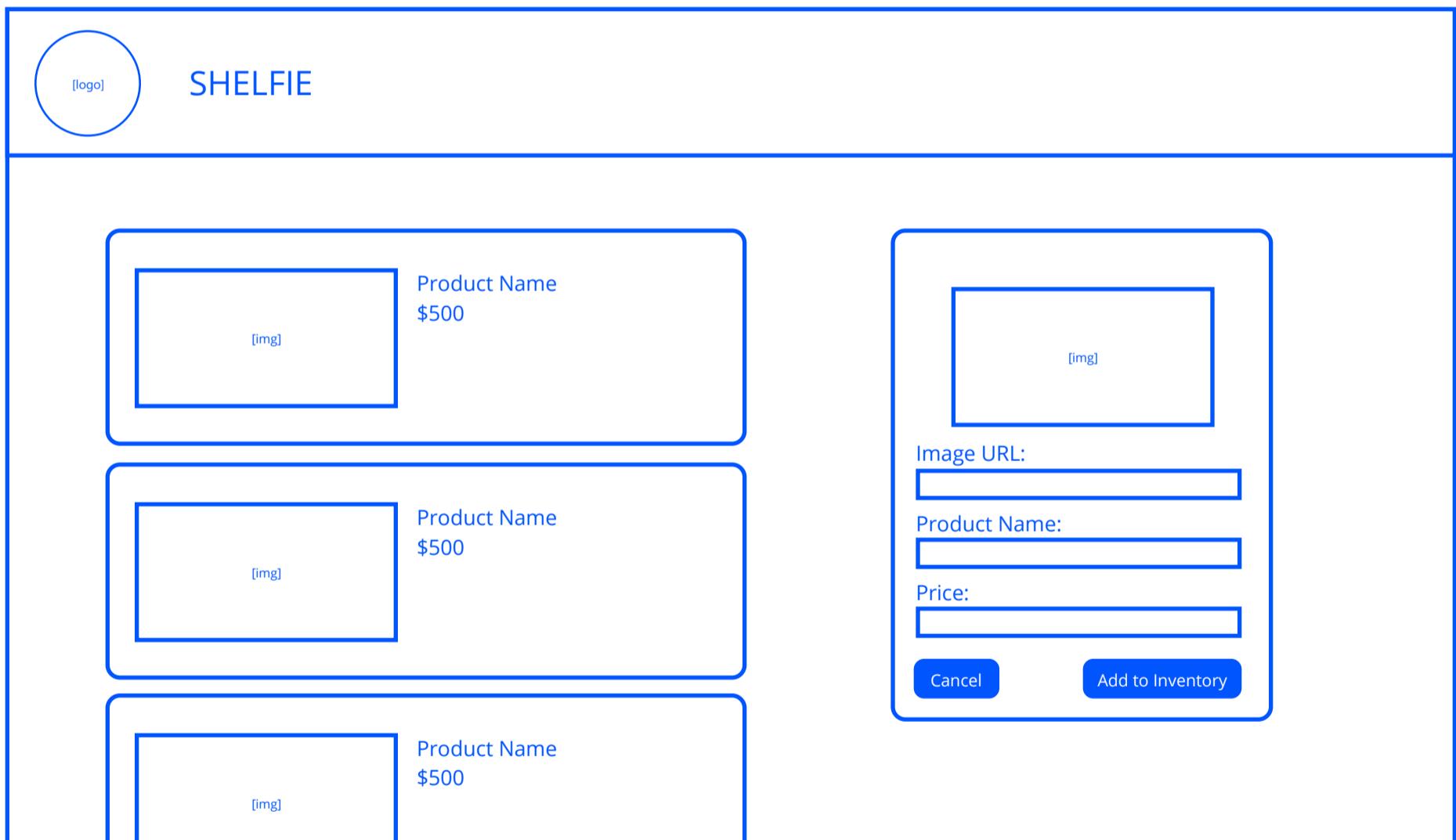
VIEWS

What views do you need to create to meet each feature in your app? Draw below.

How will the user get to each view? Draw below.

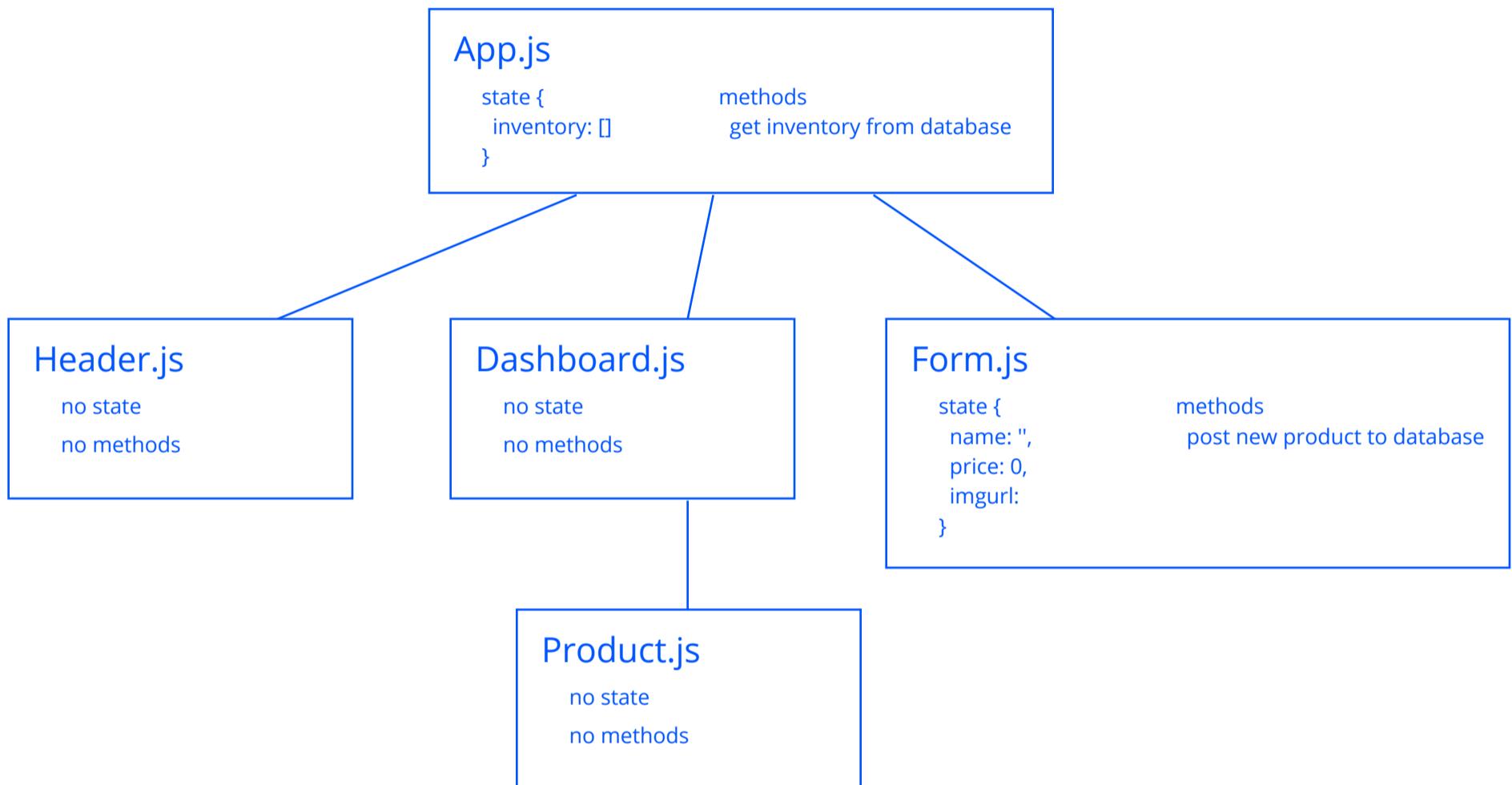
Revisit this regularly. Simplify each time. Focus on the user.

Draw each view and map out how the user will get to each of them.



COMPONENT TREE

Create a component tree for your app with all the components and their initial state. Make a connection to each endpoint it needs to use to get its data.



ENDPOINTS

List the url, REST method, and a sample of the data being sent or received for every endpoint you need.

GET '/api/inventory'

Receive: Nothing

Send: [

```
{id: 1, name: 'Example', price: 700, img: 'http://example.com/image.png'},  
{id: 2, name: 'Different', price: 80, img: 'http://example.com/image2.png'}  
]
```

POST '/api/product'

Receive: req.body {name: 'New name', price: 890, img: 'http://example.com/newimage.png'}

Send: Status 200

SCHEMAS

Create your schema(s) here. Make sure to include data types and any foreign key constraints.

products

<i>column name</i>	<i>datatype</i>	<i>constraints</i>
id	serial	primary key
name	varchar(25)	
price	integer	
img	text	

FEATURE PROGRESS TRACKER

List the features below

Component file created
Create dummy data
for component
Test component is
functioning with dummy data
Build endpoint for data
Test endpoint with dummy data
Connect endpoint with Postman
Test component to frontend
with actual data
Component styled