

JavaScript PERN Stack Project

To get started, create a copy of this document. This is one of the assignments required for your Blue Badge Project and it is highly recommended that you use this to plan as much as possible before you get started on the project. It will be a helpful resource for you.

ARTISAN GOODS - eCommerce storefront

Requirements:

- When is the application due?
April 29th @ 1pm
Team deadline: Tuesday, April 27th EOD
This will allow us time to tweak the schedule depending on where we run into issues. It also provides us with time to really dive into our stretch goals once we are sure the MVP is set up properly and in working order.
- What are the requirements for the application to be considered Minimum Viable Product(MVP)?

Server:

- Have AT LEAST two (2) tables in your database
 - One (1) of those tables must be a 'users' table that stores user information (CUSTOMERS)
 - One (1) of those tables must implement full CRUD (Create, Read, Update, Delete) functionality
- Must support session validation with JWT
- Must encrypt sensitive material such as passwords
- Must include AT LEAST 6 API endpoints
- Must be deployed with Heroku

Client:

- Must be written in React
- Must utilize custom styling in ALL VIEWS
- Add form validation to user signup/login:
 - An email address should look like 'test@test.com',
 - If a username is used instead of an email address, the username should be at least 4 or more characters and needs to include at least one (1) number or special character.

- Password should include 5 or more characters
 - Must connect with the server's deployed URL
 - Must display the functionality of the minimum 6 endpoints in the server
 - Must be deployed
- What technology will be used for project? (e.g. HTML5, CSS3, Angular, JS, etc.)
 - HTML
 - CSS
 - JS
 - React
 - PostgreSQL
 - Express
 - Node
 - REST API's and/or creating API endpoints
 - Connecting server and client
 - Team Github
 - Sequelize
 - Heroku
- Have you created two (2) GitHub repositories for the application?
 - Server repo URL: <https://github.com/rachelclayton26/ecommerce-server>
 - Client repo URL: <https://github.com/rachelclayton26/ecommerce-client>

- Does this need to be deployed and where?

Yes - Heroku (Client & Server)

- About Application: (What does it do, what is its goal?)

Goal:

For the customer:

- Make it easy to browse and shop for products on www.team4.com
- Facilitate e-commerce transactions (mock-only: no built in checkout, just shopping cart)
- Being able to list a product price, description, number available, and image.
- Client register, sign-in, and logout that allows users to add product to cart and come back to that cart later.
- Token and encryption elements

For the admin:

- Login for administrative control (validated)
- Add, update and delete product descriptions, img, etc.
- Token and encryption elements
- Stretch (add products that are not yet deployed).

How functionality supports that goal for users:

For the customer:

- Splash page that leads directly into product page on scroll (easy ui)
- Shopping cart that the product can be added and deleted from
- Click event on product image that leads to pop-up or page of product with description, "add to cart button" and product/number availability (ex. "Only 5 left in stock!", "Sold out")

For the admin:

- Hidden Login page
- Add, update, delete buttons for products
- Sample shopping cart for testing (?)

- Target Audience: (Who is the primary audience?)

Users:

- Currently, we are thinking that our product category will be moderately priced (\$-\$\$\$\$) home-goods. Therefore, we are planning to initially target millennials in their late twenties, early thirties, who are DINKS (dual income no kids), and appreciate modern, clean, decor. Our product may end up changing, based on image/description availability, but our target audience will likely stay the same during iteration.

Planning:

Logic and Functionality

Take some time to think about what is going to happen in your project. What logic is required to meet the bare minimum of your project requirements? Once you have achieved those, what would you like to see your project do that is a cherry on top of what you have already created? Where in your project does this need to take place? What moving pieces do you need?

- Server:

- MVP
 - User:
 - Signup
 - Login
 - Error handling
 - Inventory
 - title
 - Description
 - image

- price
- Keyword/tag
- count
- Stretch Goals
 - Shopping cart
 - Tags/collections
 -

- Client:

- MVP
 - User:
 - Signup
 - Login
 - Storefront Main
 - Storefront Item Detail
 - description
 - pic
 - fave/save

- add to cart
-
- Stretch Goals
 - Shopping Cart
 - tags/collections
 - Personalization
 - Cart abandonment campaign
 - communication opt in

Tables and Endpoints

Think about what your database is going to look like. How many pieces of data should your table expect? What dataTypes are you working with?

Table 1: User

Model	Controller/Endpoints
<ul style="list-style-type: none">• Email• Password• FirstName• LastName• Admin• id•	<ul style="list-style-type: none">• router.GET('../user')• router.GET('../user/:userid')• router.POST('../user')• router.PUT('../user/:userid')• router.DETELETE('../user:userid')• Logout••••

Table 2: Shop (Create, Read, Update, Delete)

Model	Controller/Endpoints
<ul style="list-style-type: none">• title• description• price• tag or keyword• id•	<ul style="list-style-type: none">• router.GET('../shop')• router.GET('../shop/id')• router.POST('../shop')• router.POST('../shop/id)//admin• router.PUT('../shop')• router.DETELETE('../shop/id)//admin•

Table 3: Cart

Model	Controller/Endpoints
<ul style="list-style-type: none"> • customer • Owner • Products • • • • 	<ul style="list-style-type: none"> • router.GET(• router.POST(• router.PUT(• router.DELETE(

Trello

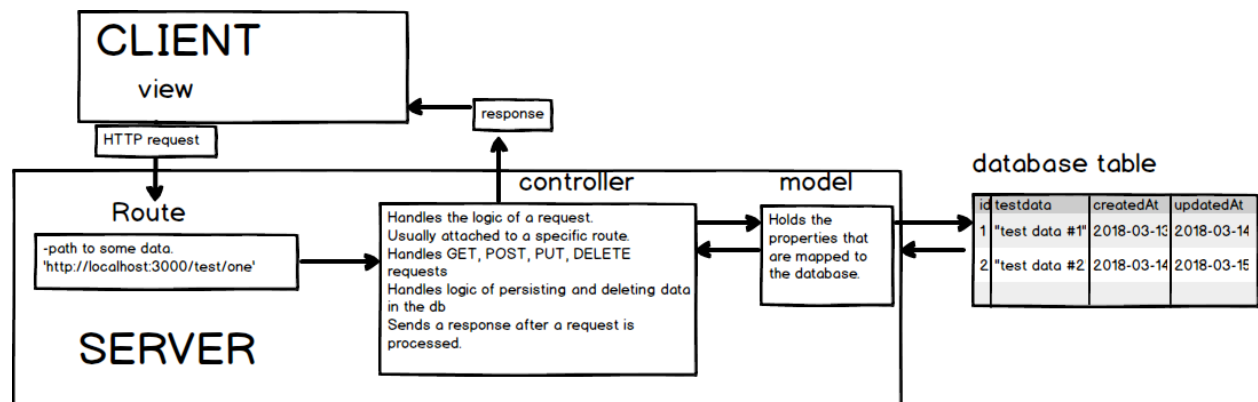
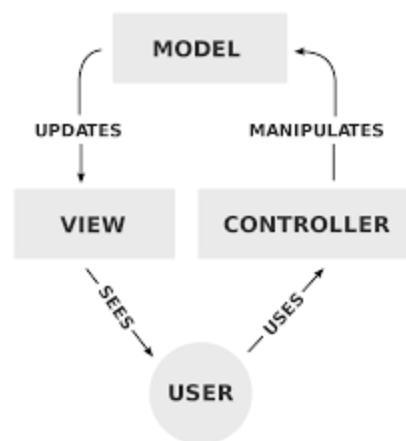
Trello is a great tool to help you organize yourself, a team, or even your family! In software development, we like to use Trello to create and keep track of 'tickets': cards with descriptions, ideas, or step-by-step logic to build out our applications. While this is not a requirement for your Blue Badge Project, it is a helpful way to manage your time, ideas, and keep track of your concepts to ensure your success during your first PERN Stack Project.

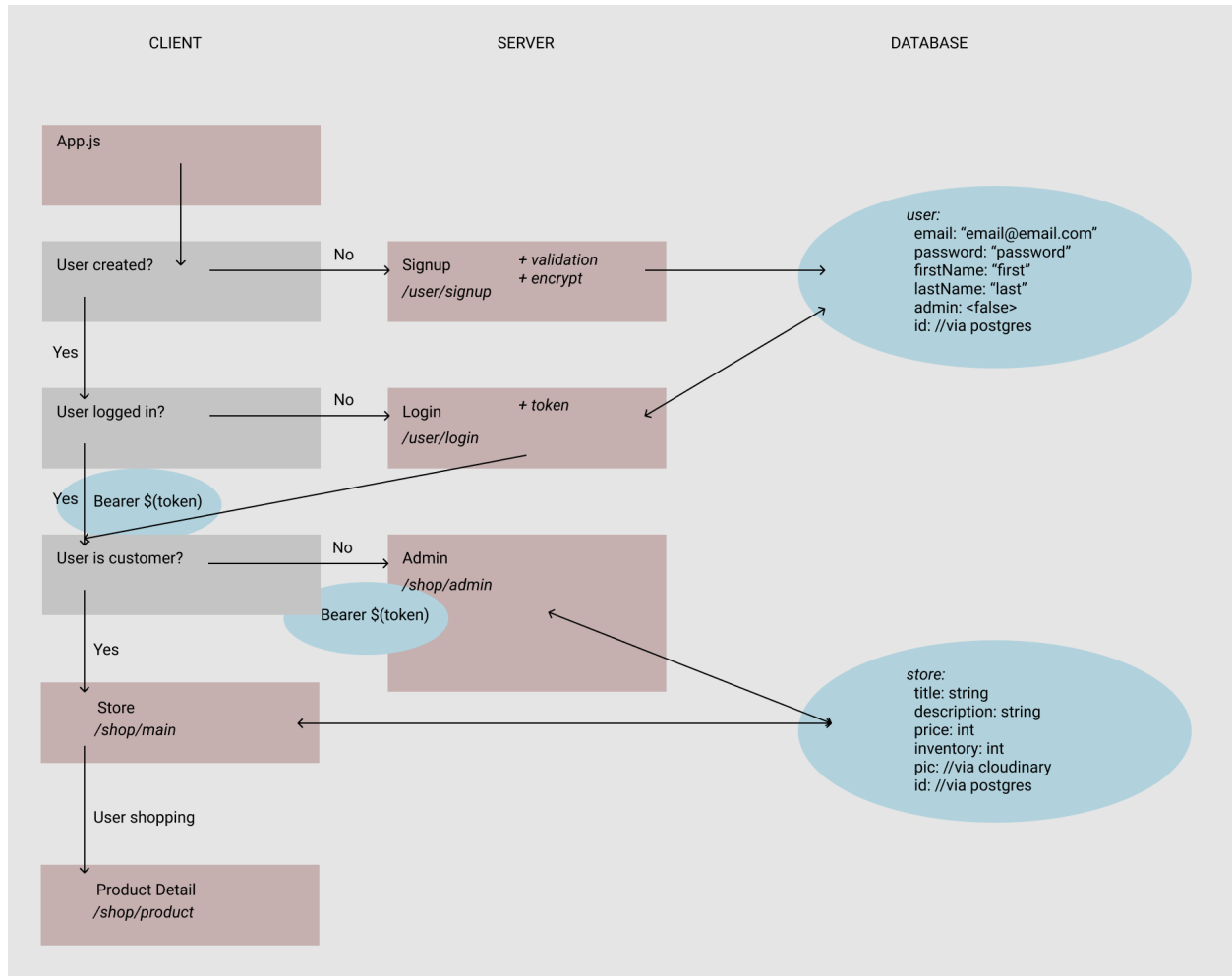
[Click here to get started!](#)

[Here is an example of a Trello board for a Blue Badge Project.](#)

Data Flow

When planning a project, we also need to consider what data needs to be available in certain components. Where do you need to declare your variable and which components feed into each other so that the value of the variable is accessible when you click here or navigate to this part of your project? Building out the data flow of your project will also be helpful when building out your folder structure. Take the time to draw out where the structure of your project, where you will declare certain variables and build out pieces of logic, and trace the flow of where they will be utilized. Remember that React components look similar to a tree structure.





Styling:

This is where you draw out a blueprint or a skeleton of the visual layout of the project. This is also known as the UI (User Interface). This can be drawn out with paper and pencil, whiteboards, and/or tools such as [sketch](#), [xd adobe](#), or [balsamiq](#). Consider how the components will fit into each and what data needs to be passed between components. This is also a great place to think about adding a [style guide](#) and words or images that you are using for inspiration.

- Color Scheme:

TBD

- neutral
- minimal design complexity - let the products be the feature & color

- Style Guide:

font family: TBD

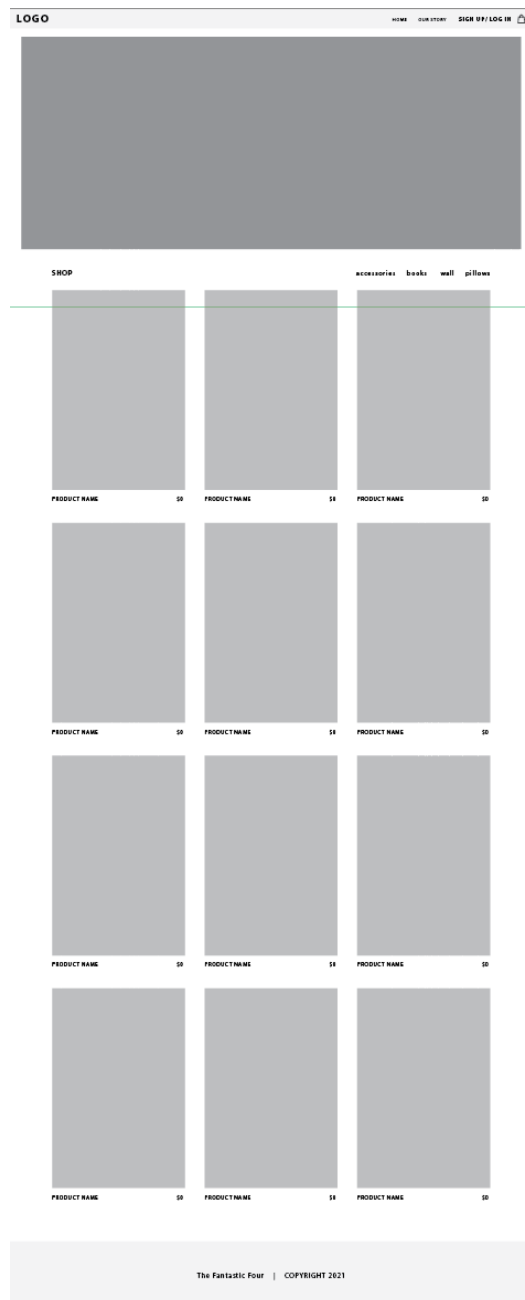
UI frameworks:

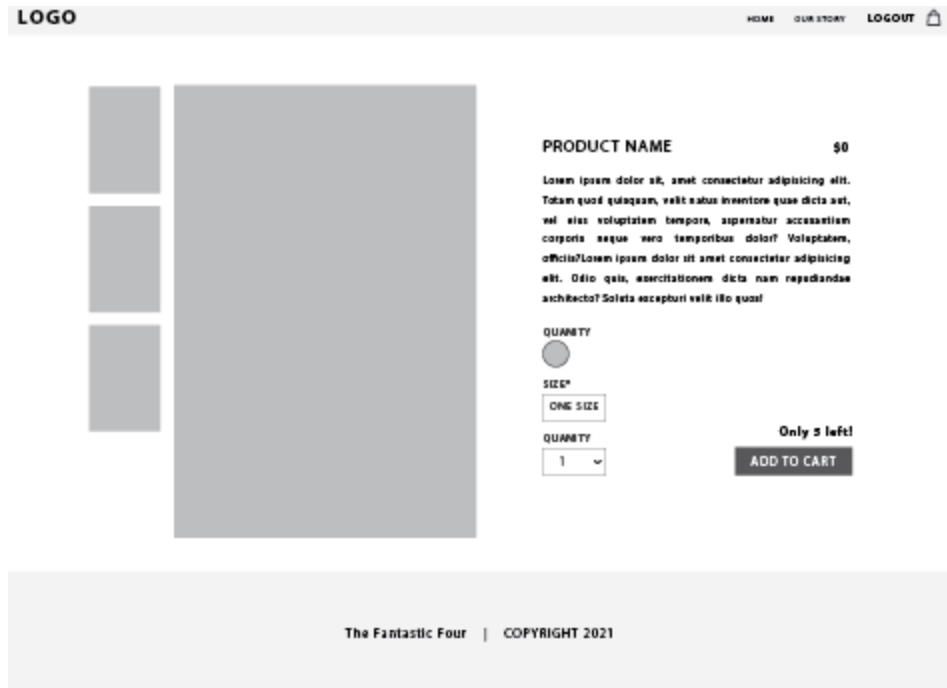
Custom styling:

3x4 grid of products

Wireframe:

Customer View:





Admin Portal:





SHOP

accessories books wall pillows



EDIT
DELETE

EDIT
DELETE

PRODUCT NAME \$0

PRODUCT NAME \$0

PRODUCT NAME \$0

EDIT
DELETE

EDIT
DELETE

EDIT
DELETE

PRODUCT NAME \$0

PRODUCT NAME \$0

PRODUCT NAME \$0

EDIT
DELETE

EDIT
DELETE

EDIT
DELETE

PRODUCT NAME \$0

PRODUCT NAME \$0

PRODUCT NAME \$0

EDIT
DELETE

EDIT
DELETE

EDIT
DELETE

PRODUCT NAME \$0

PRODUCT NAME \$0

PRODUCT NAME \$0

Schedule:

Create a schedule for what you will complete each day for the project. The table below is a guideline. It is not necessary to specify each day's work/logic. Feel free to estimate your time.

			April 22	April 23	<i>April 24</i>	<i>April 25</i>
			-planning -repos -start endpoints	Workshops 9-5 -finish servers	-finish servers -start client	-client
April 26	April 27	April 28	April 29	April 30	Day	Day
deploy server -client MVP complete	deploy client -debug -team due date	-final changes & stretch goals	Demo day!			
Day	Day	Day	Day	Day	Day	Day