



Bubbly

Bubblify

A major catastrophe has just happened! All the key players in the bubble industry have been relying on the same manufacturer which just recently went through a really bad lawsuit which resulted in them going bankrupt! It is already May and the summer is almost here and kids want their bubbles! A new emerging company called **Bubblify** has decided to jump on this chance and become the new big player in the bubble industry. They want their business to go mainly through their website, which has not been made yet... That's where you come in! You are going to build a website for them before the summer season begins. Please, do not disappoint them nor all the soon-to-be bubbleless children...

TypeScript

All React code written in this course should use TypeScript and this assignment is no exception. You can feel free to use Create React App to scaffold your template.

<https://create-react-app.dev/docs/adding-typescript/>

Assignment description

All the functionality in the assignment is enlisted here below:

- **(10%)** Implement a site where you can look at all the available bubble products (e.g. <http://localhost:3000/bubbles>)
- **(10%)** All bubble products contain a link to a detailed page for each product, where extra information is provided (e.g. <http://localhost:3000/bubbles/1>)
- **(10%)** Implement a site where you can look at different bundle products (e.g. <http://localhost:3000/bundles>). Bundle products are combined of two or more products
- **(10%)** Implement a site where you can look at information on the place **Bubblify** (e.g. <http://localhost:3000/about>)
- **(10%)** Implement a site where you can look at all the products you have added to the cart. Cart items should be stored either in **localStorage** or cookie and retrieved when the user comes back (e.g. <http://localhost:3000/cart>)

- **(5%)** The navigation bar should contain a logo (*you can use any image you find suitable*) and four links
 - Products
 - Bundles
 - About us
 - Cart
- **(35%)** A user needs to go through the following steps when buying products:
 1. Select one or more products from the product section or select previous order which can be retrieved at (<http://localhost:3500/orders/:telephone>)
 2. When a selection is finished, he gets an option to go the checkout. The checkout should also be available in the cart
 3. In the first step of checkout the user needs to decide if he wants to arrange a store-pickup or have it delivered
 4. If he decides to have it delivered, the next step will be to input the following information: **name, address, city, telephone** and **postal code**
 5. If he decides to arrange a store-pickup, the next step will be to input only the **name** and **telephone**
 6. At the final step he gets a review of his order and if he is satisfied he can click a button to confirm
 7. When the confirm button is clicked he is redirected to a confirmation site, which should tell him that all was successful. The order should be stored in the server (<http://localhost:3500/orders> [POST]). You choose on what format the order should be which you send via request body.
 8. Information about the user should be stored in either cookie or localStorage
 9. Forms should be properly validated
- **(5%)** Components which receive props should be **PropTypes** set
- **(5%)** All **PropTypes** should be documented

Resources

All packages are available.

Other

A **server** is provided and can be downloaded in the **Assignment** section in **Canvas**. This server should provide you with all the functionality you need, but if you want to change it, feel free. If you change the server code, include the new server code with your submission and update the **README.md** accordingly.

Submission

A single compressed file (*.zip, *.rar) should be submitted in **Canvas. node_modules/** should be excluded.