Noroff
School of technology and digital media

Project Exam 2 Resit

Goal

To take the skills learned over the last two years and take on an extensive project where the finished product should reflect the candidate's general development capabilities, in addition to visual and technical skills.

Brief

The final submission must have the following:

- A Gantt chart planning the project
- A style guide
- An Adobe XD (or other UI tool) prototype
- Use a CSS pre-processor
- Use React
- Follow BEM
- A fully working website that fulfils the brief

The Project

You've been tasked with designing and developing a new **video game store** website called Bits & Bots. The owners want only registered users to be able to browse the site. The site must be responsive and work on all device sizes.

Requirements

Landing page

The landing page must include either a fullscreen video or slide show. Make sure you are allowed to use the video or images you source for this.

The site requires registration and there must be two tabs in the middle of this page:

1. Login
2. Register

The register tab must hold a form with email address and password inputs. Submitting the form must validate the inputs and write the values to localStorage.

The login tab form must also contain email and password inputs and on submit check the values against those stored in localStorage. If the values match, the user will be redirected to the browse page, otherwise an “Incorrect username or password” message must be displayed.

Browse page

When first loading the browse page, an API call will be made to load a list of games. A loading indicator must be displayed while the API call is in progress.

Once the call is complete, the page must display the list of games as items in columns and rows. The title and image of each games must be displayed.

At the top of the page will be a list of genres as tags, tabs or menu items. Clicking on a genre will load games of that specific genre.

Each item must have a *more info* or *details* link. Clicking this will take the user to the details page with the id of the game in the url, e.g. /details/5

Somewhere on each game item there must be a *favourite* or *add-to*-cart icon button. Clicking on the icon must add the game info in an object to an array in localStorage. This will act as the site’s shopping cart. Clicking it again will remove it from the cart. The icon must indicate whether or not it is in the cart.

At the top of the browse page there must be a *view cart* button which takes the user to the cart

page.

Details page

This page must retrieve the id from the url and make an API call to fetch the details for this game using the id.

The student can decide which fields to display from the API call result.

The page must also include the *favourite/add-to-cart* icon and functionality.

Cart page

The cart page must display the list of games stored in the localStorage cart. All info for this list must be retrieved from the cart. A remove button must exist on each item. Clicking this will remove the game from the list and from localStorage.

If there are no items in the cart the page must display a relevant message.

A *check out* button on this page will take the user to the check out page. If there are no items in the

cart this button must be disabled.

Check out page

This page will have a form with inputs for the user's name, address and credit card details. Relevant validation must exist on each input.

Above the form the page must display the number of items in the cart.

If the input values all pass validation, the submit button must open a modal to confirm payment.

Once confirmed, the cart array will be emptied and the user redirected to the browse page.

Common page elements

- A log out button which will clear all information stored in localStorage and redirect the user to the landing page
- Suggested: A breadcrumb element to make navigation easier

API

Deploy your own API. You can use Strapi, WordPress API or another Headless CMS.

Component library

Consider using a component library like Bootstrap for your project.

Recommended Process

- Week 1 and 2: Planning and Design
- Week 3 to 6: Coding
- Week 7: Bug Fixing

Level 1 Process

1. Start by planning out the project. Develop a Gantt chart and a functional specification.
2. Next begin your research of likely users and browse competitor websites for ideas.
3. Create a style guide.
4. Build a prototype in a UI tool like XD, Sketch or Framer-X.
5. Create a repo on GitHub. Be sure to commit your code regularly.
6. Start coding. Ensure you use a CSS pre-processor, follow the BEM methodology, and use React to build the project.
7. Once you've finished development, start looking for bugs and ensure the site functions well on all viewports.

Submission

End of week 7:

Report with a link to your GitHub repo. Make sure the README file in your repo clearly explains how to run the project. Your report must include a functional specification, Gantt chart, style guide and prototype.

© Noroff