Web Application
& Interface Design
Presentation

Overview

Websites are the front door to any business, and especially critical for a restaurant. 90% of customers that visit a restaurant have already researched it online prior to dining there, more than any other type of business. With so much interest and competition, it's extremely important to make the right first impression. This is why the elegant, fine-dining establishment, Café Fausse, has reached out to your team for a brand-new website. They know you can deliver a product that is in line with their own high standards. It looks like you will need to pull out all the stops to impress them!

To do this, you will be developing a complete web application to meet specific software requirements. You are not expected to have carried out significant testing (testing approaches are to be covered in a later concentration), but you should have a complete development version that at least addresses all software requirements.

While the web application can be completed as part of a group of up to no more than 3 people, every student will be submitting their deliverable **individually**. This deliverable includes a recorded presentation in which the student alone must demonstrate the functionality of their web application.

Learning Outcomes

When completed successfully, this project will enable you to:

- Develop working websites using React and JSX, and Flask
- Translate customer requirements into a working web application
- Carry out high quality UI and UX design
- Develop backend logic and a backend database using Flask and PostgreSQL
- Implement client-side forms with React and form handling with Flask
- Utilize AI tooling for rapid and high quality software development

Project Description

You have been provided both with a copy of the:

- software requirements specification for this project, and
- a collection of images for you to use.

While you can fully hand code this project if you wish, you are highly encouraged to utilize leading AI code generation models/AI IDEs to assist in rapidly producing your Web application, being sure to describe in broad terms how you made use of them. Here are some examples of very useful AI tools you may wish to consider. You will be graded on the quality and functionality of the application and how well it meets the requirements—no given proportion of the code is required to be hand coded.

Hint: Start by loading the software requirements specification into your AI model of choice, and then enhance or modify your code using an AI-based IDE, such as Cursor or Windsurf.

You have been tasked with crafting a website from scratch using your knowledge of **HTML, CSS**, and **JavaScript**. Restaurants are dependent upon intuitive interfaces and impressive visuals. This is where **React** will come into play. React is a flexible and efficient JavaScript library used to build user interfaces. It accomplishes this by combining small code elements, or “components”, into much larger designs. With an easy-to-learn approach and a wide array of resources and support, React can provide the ability to scale and reuse components across much larger projects. It’s a handy tool to have in your kit. To get started, visit the React tutorial page here.

Work through the tutorials to get familiar with the React library and how to integrate it. Once you are comfortable, your team can begin planning the restaurant site.

First, consider the fundamental functions the customer wants to see on the site:

- Contact information: Address, phone number, hours
- Their menu, broken up by categories
- An “About Us” page highlighting the owners
- An email sign-up for a newsletter
- A photo gallery page, and judicious use of photos throughout the site
- Something showcasing their awards and positive reviews
- An ability to make a table reservation via the Web site

You will also want to provide a constant thematic look and feel by applying a consistent CSS for the entire site. You can either build this directly or utilize CSS-in-JS to code through the React scripts. Your CSS implementation should take either a Flexbox, or Grid approach to design. For more information, check here.

In addition to needing a new web site providing a great web presence, the owner of Café Fausse is tired of fielding so many calls for available tables. What they need is a web-based system that provides the capability to accept reservations for customers. This system should be able to select specific timeslots, input the number of guests, assign a name to the reservation, require an email address and an optional phone number and store this information in the system. It should also check to ensure the timeslot is not fully occupied.

To address the reservation system requested by the owner of Café Fausse, you will need to build both a front-end interface using forms, and a back-end SQL database that stores the reservations. This should be accomplished with a combination of a React front-end form, and using Flask/Python and PostgreSQL for back-end logic and database respectively. Your PostgreSQL database should have, at a minimum, a table for **Reservations** and **Customers**. **Customers** should have a "Customer ID", "*Customer Name*", "Customer Email", "*Phone Number*" and "Newsletter Signup". **Reservations** should have a "Reservation ID", "Customer ID", "*Time Slot*" (date and time), and "*Table Number*". The back-end logic should take the customer's information, add it to the customer table, then assign a random table for the chosen time slot (assume there are 30 tables total). Once completed, it should confirm to the user of a successful reservation. If all seats are taken for that time slot, the system should send back a message that the customer needs to pick another time.

In total, your website should have a minimum of five pages:

- Main (index) page
- Menu
- Reservations
- About us
- Gallery

The additional elements listed above may be present on one of these pages or others. The email sign-up should utilize **form** elements to provide this functionality and include basic verification on the input fields, and also store sign-ups in a backend database. Once you are satisfied with the look and feel, test your site to ensure all links work and the behavior is intended. Verify that it can operate on multiple browsers and platforms. Consider using a mobile emulator tool, such as Chrome's mobile simulator testing tool.

Note: *Your team can use additional photographs/images on the web site but ensure that they are royalty free. You can also or alternatively make use of additional AI-generated images.*

The final site and all of its components should be hosted either 1) locally (localhost) or 2) on an accessible staging server. In a submitted screenshare demonstration, you will demonstrate all functionality of your Web application.

Tips & Resources

- AI Code Generation Tools
- Creating a React project in Windows
- CSS Syntax reference
- CSS Grid and Flexbox
- Flask and React
- Responsive Design
- Responsive Design Best Practices
- Python and PostreSQL

Presentation Requirements

- The presentation should be submitted via a recording where you are present and visible on-screen AND your presentation is recorded on your screen. You can find more information on how to record your video presentation here.
- The demo presentation should last around 5-10 minutes and demonstrate all site functionality, including:
  - Each of the five (or more) pages and navigation between them
  - The email signup for the newsletter
  - A correctly functioning reservations system
  - The correct effect of reservations and newsletter signup on the state of the backend database
  - Discussion of the implementation decisions you have made
- In your presentation, you must:
  - Present a government-issued ID to the camera so your name and picture are clearly visible and legible.
  - State your name.
- **Please do not "Invite People" in order to share your presentation—there are multiple graders who could potentially be grading your project, and they do not have access to the projects@quantic.edu account.**

Submission Guidelines

Every student must submit **individually**. Your submission should include both 1) a link to your recorded demo presentation and 2) a document containing the following:

- A link to a private GitHub repository containing all source code used for your site (frontend and backend). You must add "quantic-grader" as a collaborator to your private repo. Add this collaborator using the repo's Settings > Collaborators > Add people.

    - Include a README file in the repo that describes your solution and how to run it locally
- A summary of any AI tooling you used in developing your solution and how generally you used it (i.e., what worked well.)
- *Optional*: a link to the staging version (if you have created this) or indicate that you only run it locally

To submit your project, please click on the "Submit Project" button on your dashboard and follow the steps provided in the Google Form. If you have chosen to work with a group on the web application, you will also be prompted to upload the final page of your Group Project Agreement, which must be completed and signed by all group members. Please reach out to msse+projects@quantic.edu if you have any questions. Project grading typically takes about 3-4 weeks to complete after the submission due date. There is no score penalty for projects submitted after the due date, however grading may be delayed.

Plagiarism Policy

Here at Quantic, we believe that learning is best accomplished by “doing”—this ethos underpinned the design of our active learning platform, and it likewise informs our approach to the completion of projects and presentations for our degree programs. We expect that all of our graduates will be able to deploy the concepts and skills they’ve learned over the course of their degree, whether in the workplace or in pursuit of personal goals, and so it is in our students’ best interest that these assignments be completed solely through their own efforts with academic integrity.

Quantic takes academic integrity very seriously—we define plagiarism as: “Knowingly representing the work of others as one’s own, engaging in any acts of plagiarism, or referencing the works of others without appropriate citation.” This includes both misusing or not using proper citations for the works referenced, and submitting someone else’s work as your own. Quantic monitors all submissions for instances of plagiarism and all plagiarism, even unintentional, is considered a conduct violation. If you’re still not sure about what constitutes plagiarism, check out this two-minute presentation by our librarian, Kristina. It is important to be conscientious when citing your sources. When in doubt, **cite**! Kristina outlines the basics of best citation practices in this one-minute video. You can also find more about our plagiarism policy here.

Project Rubric

Scores 2 and above are considered passing. Students who receive a 1 or 0 will not get credit for the assignment and must revise and resubmit to receive a passing grade.

Score	Description
5	Addresses ALL of the project requirements, including: ● The minimum five pages of the Web site have been built using React and JSX ● All of the requirements requested in the SRS have been implemented ● The Web site maintains good appearance and evidences excellent UI and UX design ● Appropriate use of Flexbox or Grid approaches has been made to achieve a high quality UX ● Where forms are needed to meet user requirements, these have been correctly implemented and are working ● A back-end Flask app and database correctly integrated with the React front-end and meeting the requirements for the reservation and newsletter signup system ● A document outlines what AI code generation tools have been used and how
4	Addresses MOST of the project requirements, including: ● At least four pages of the Web site have been built using React and JSX ● Most of the fundamental functions requested in the SRS have been implemented ● The Web site maintains good appearance and evidences good UI and UX design ● Appropriate use of Flexbox or Grid approaches has been made to achieve a high quality UX ● Where forms are needed to meet user requirements, these have been correctly implemented and are working ● A back-end Flask app and database correctly integrated with the React front-end and meeting the requirements for the reservation and newsletter signup system

	● A document outlines what AI code generation tools have been used and how
3	Addresses SOME of the project requirements, including: ● Three or more of the pages of the Web site have been built using React and JSX ● Some of the fundamental functions requested in the SRS have been implemented ● The Web site maintains good appearance and evidences good UI and UX design ● Appropriate use of Flexbox or Grid approaches has been made to achieve a high quality UX ● Where forms are needed to meet user requirements, these have been correctly implemented and are working ● Where forms are needed to meet user requirements, these have been correctly implemented and are working ● A back-end Flask app and database correctly integrated with the React front-end and meeting most of the requirements for the reservation and newsletter signup system ● A document outlines what AI code generation tools have been used and how
2	Addresses FEW of the project requirements, including: ● Two or more of the minimum five pages of the Web site have been built using React and JSX ● Few of the fundamental functions requested in the SRS have been implemented ● The Web site maintains average appearance and evidences limited UI and UX design ● Appropriate use of Flexbox or Grid approaches has been made to achieve a high quality UX ● Where forms are needed to meet user requirements, these have been correctly implemented and are partially working ● A back-end Flask app and database correctly integrated with the React front-end and meeting few of the requirements for the reservation and newsletter signup system ● A document outlines what AI code generation tools have been used and how

1	Addresses the project but MOST of the project requirements are missing, including: ● Less than two of the pages of the Web site have been correctly built using React and JSX ● Most of the fundamental functions requested in the SRS have not been implemented ● Poor UI and UX design is evidenced ● Appropriate use of Box Flexbox or Grid approaches has not been made to achieve a high quality UX ● Where forms are needed to meet user requirements, these have not been correctly implemented ● A back-end Flask app and database correctly integrated with the React front-end has not been achieved ● A document outlining what AI code generation tools have been used and how is not included
0	● The student either did not complete the assignment, plagiarized all or part of the assignment, or completely failed to address the project requirements.