Project Name: X-Scape

Group 2: Ethan Wang, Stephanie Wang, Arianna Khan, Cici Zhao, Yifei Tao

Prototype and Testing Protocol

Will you create an interactive prototype before you begin coding (e.g. using Figma), or will you work from static wireframes?

We will be creating an interactive prototype using Figma and advancing our static low-fidelity wireframe to interactive high-fidelity wireframes. As a group, we acknowledged the importance of making an interactive prototype to maintain clarity for which pages link to what and what the website pathways should entail. Having an interactive prototype can ensure everyone is on the same page and that the final product prototype will translate directly to our actual interactive web application. Currently, we are collectively working towards design guidelines for our website and have decided on the look and feel of the site(color palette, typography, etc.) as well as the usability and navigation aspects for users.

Link to our Figma file

What are the acceptance tests that your team will perform before beginning user testing? Describe the acceptance testing process for at least two key features, including which results would indicate a successful test and which results would indicate a failed test.

	Feature 1: Search and Filter	**Feature 2: Compare**	**Feature 3: Quiz**
Requirement Specification	Users should be able to search and filter content on the web app	Section that includes a table that compares different components including features, type of car, and carbon emissions.	Users could participate in a quiz to discover the three most suitable models that align with their specific requirements.
Positive Test Scenario	Successful Product Search: ● Test Scenario: A user enters a valid search query and	Successful Comparison: ● Test Scenario: A user enters a search query for a car in each compare table	Quiz Completion ● Test Scenario: A user successfully completes the car recommendation

	receives search results. • Expected Result: The user sees a list of products matching the search query and can click on the results to view details. • Failed Test: The user receives no search results, or the results are inaccurate.	• Expected Result: The user sees the car cards next to each other with car stats displayed for comparison • Failed Test: the user doesn't see any car displayed, the results are inaccurate, or none of the compare features are visible	quiz by answering all questions. • Expected Result: The user is presented with a list of recommended car models based on their quiz responses. • Failed Test: The quiz does not provide any recommendations , or the recommendations do not match the user's responses.
Negative Test Scenario	Invalid Search Query: • Test Scenario: A user enters an invalid or empty search query. • Expected Result: The system displays an error message, guiding the user to enter a valid search query. • Failed Test: The system either returns results for an invalid query or does not display an error message for an empty query.	Invalid Compare Table: • Test Scenario: A user has two cars in mind to test, selects them from the drop down menu, and applies search. • Expected Result: The system prompts user with a loaded table with information • Failed Test: The system does not display a table with information or user cannot apply search	Incomplete Quiz: • Test Scenario: A user attempts to skip questions or submits an incomplete quiz. • Expected Result: The system prompts the user to complete any unanswered questions before generating recommendations • Failed Test: The system allows the user to submit an incomplete quiz, or it generates recommendations

			without all necessary input
Testing Cases	• Perform a search for specific car model • Apply filters to refine search results • Ensure the content displayed matches the search and filter criteria	• Select 2 cars from drop down menu • Ensure the car models are displayed correctly • Verify the selected car matches the description	• Start and complete a quiz • Ensure that questions and choices are displayed correctly • Verify that the quiz submits and provides top 3 recommended cars
Expected Results	• The web app returns relevant results • Filters can be applied and accurately refine the results • The displayed content matches the search and filter criteria	• Able to clearly see the differences between the two car models they selected • Able to type out a part of the car name and find the item they are looking for from the prompted drop-down menu without difficulty or confusion	• Provides the user with three different cars that satisfy the requirements a user is looking for in a car • Returns cars that are eco-friendly
Expected Deficiency	The filtered result may take some time to generate.	The comparison table results may take some time to generate.	The quiz result may take some time to generate.
Unexpected Results	Users are having trouble locating the items they seek or encountering challenges while	Users might not be able to find the specific car make and model they have in	Users might have a hard time deciding between choices and might be unsure of a final

	navigating this functionality. There is a lack of information and text that communicates "No results found" to serve as a reminder to users that there are no cars meeting the chosen criteria.	mind. Users end up choosing the same two cars to compare.	decision.

What are the limitations of your acceptance tests? List some of the ways that your team's in-house testing environment may differ from the context in which your expected users will be interacting with the product.

1. **Different User Behavior and User Data**

   During in-house testing, we will follow predefined scripts and use sample data, which may not fully represent the diverse and dynamic behavior of actual users. For example, for the Product Search feature, real users might input a wide range of search queries(e.g., misspellings, slang, abbreviations), which the in-house tests might not account for.

2. **The Performance of Website Server**

   In a controlled in-house testing environment, the MVP could perform well, but it may not accurately reflect how the system behaves under real-world loads. When real users access the websites simultaneously, server loads could fluctuate and might cause issues. Under sudden surge in user traffic, the server might slow down or be unresponsive, and this kind of scenario is one the in-house tests may not replicate.

3. **The Difference in Operation System, Device and Browser**

   Our in-house testing environment is limited to a specific set of devices(PC vs. phones), browsers(Google Chrome), and operational systems(mac vs. Linux). However, in a real user context, users will access our MVP through a wide range of devices and browsers. For example, some users may use older versions of browsers that do not fully support the product’s features, which leads to functionality issues. This is also one context that our in-house testing may not uncover.

How will you conduct user testing?

We will conduct user testing by interviewing users regarding our idea and the functionality of our website. More specifically, we will conduct actual usability tests by sitting down and showcasing our interactive Figma prototype to individuals to determine our site's usability and intuitiveness. Each person in our group will ask at least one individual to sit down for approximately 30 minutes and use our web application.

Steps to conduct usability test:

1. Introduce to the users a brief overview of the purpose of the website and what components make up the website
2. Confirm with users for their consent to proceed and ask if they are comfortable with us writing down any notes about their experience
3. Ask users to complete 3 tasks on website to test each of our features for functionality (Quiz, Search, and Compare) and use the 'Think Aloud Method' (ask participants to talk through each step and what their thoughts are when navigating through the website)
    i. Task 1: Navigate to the quiz section and take a quiz to get your personalized quiz results
    ii. Task 2: Search up a specific type of car (e.g. 2022 Honda Civic)
    iii. Task 3: Compare 2 specific cars (e.g. 2022 Honda Civic vs. 2022 Toyota Prius)
4. Ask subjects concluding questions about our website navigation, usability and UX design to get feedback on potential areas of improvement
    a. e.g. Was it easy to navigate to the search or compare page? Was it easy to tell what our site was about? Do you like the look and feel of the website? Would you use this website or recommend it to others?
5. Thank the users for their time and express our gratitude

How will you decide which bugs to fix first?

- **Severity**: Bugs are often categorized by their severity, such as critical, major, minor, or cosmetic. Critical bugs that cause system crashes, data loss, or security vulnerabilities should typically be addressed first, as they have the most significant impact on users and the application's stability.
- **Impact**: Consider how many users are affected by a particular bug. High-impact bugs affecting a large portion of our user base should generally be given higher priority.
- **User feedback**: User-reported bugs and feature requests should be prioritized because they directly address user needs and concerns. Listening to our users and addressing their pain points can improve user satisfaction and retention.

How will you re-test the solution after the bug fixes have been completed?

1. Set up a test environment that resembles the production environment where the website will be hosted, and ensure that all bugs are fixed and the code is up-to-date.
2. Re-run the test cases that initially exposed the bugs, and make sure that the website functions well after the bug is fixed.
3. For each test case, compare the expected results with the actual results obtained from the updated websites. If the expected and actual results match, the bug fix can be considered successful for that particular case.
4. Keep exploring for unexpected issues that may not have been seen during the previous test. We would use different usage scenarios and paths to test it through the system.