CrowdTwist Design and Coding Test (Front end)

CrowdTwist Design and Coding Test

Welcome to the CrowdTwist Design and Coding Test!

Please read the following instructions carefully before beginning the test.

1. Solutions are judged primarily on correctness, but design, clarity, efficiency, and documentation are all taken into account.
2. All design and coding exercises are to be written as if you were on the job. Document as if you were working on the CrowdTwist codebase. Unit test as if you were working on the CrowdTwist codebase. Code as if you were working on the CrowdTwist codebase, and so on. If you write the coding exercise on the back of a napkin and take a picture for your submission, it had better be good!
3. Coding exercises must use standard, operating system neutral libraries for the programming language.
4. Provide solutions as a .zip, .tar, or .tar.gz.
5. Document all assumptions. If you are uncertain what is being asked, use your best judgement and document your assumptions.
6. These exercises should generally be completed within four hours.

Design

Problem #1 - OOP Analysis and Design Exercise

At CrowdTwist, we look for engineers that possess a strong ability to absorb and organize data in both traditional and unorthodox perspectives.

Design an object oriented system modeling **one** the following categories:

- Computers
- Phones
- Chairs
- Animals
- Shapes

You're design should consist of 5-10 classes and define:

- Public, protected and private member variables and methods
- Polymorphism
- Inheritance, abstract classes, interfaces, etc. (where appropriate)

In this design exercise we are looking for extensibility, reusability, and modularity of your design. At first this may seem like a trivial exercise, but it's not (hint: the categories are in decreasing difficulty). Give it some serious thought and consideration. A UML class diagram is preferred, but not required. If you're not familiar with UML, use you best judgment in providing the answer in a form that conveys your design well.

Coding

Problem #2 - Coding Exercise

Design and develop a survey creation tool. The survey creation tool contains a form which allows users to create N number of surveys. Each survey has a title, description, the number of points a user will receive for completing the survey, and one to many questions that are part of the survey. A user can enter an arbitrary amount of questions in the survey form. The form shall validate each of these inputs and produces a JSON object containing key-value pairs of form field identifiers to their respectful values. Handling server requests is **NOT** necessary.

The following is the JSON representation for an example survey survey containing two questions:

```
[{
 title: “What do you like?”,
 pointValue: 35,
 description: “Sample survey to handle gathering basic user info”,
 questions: [{
  title: “What is your favorite color”
 }, {
  title: “What is your favorite sport?”
 }]
}]
```

Validation

The following rules should be enforced on the client-side, meaning that a user filling out the form should not be able to submit it (or in this case generate the JSON object) with invalid form values.

- title - A string containing less than 100 characters
- pointValue - A positive integer
- description - A string containing less than 500 characters
- questions - A variable list of question objects containing titles. Survey's must contain at least one question.

Rules

- Feel free to use any framework of choice (ie. Backbone, Angular, Ember, plain Javascript)
- No third party libraries allowed (ie. Backbone Forms, jQuery Form Builder)

Extra Credit

Form could be pre-populated with an existing JSON object, ie. Edit mode.

Architecture

Problem #3 - White-label Architecture

Design a client-side architecture for a white-label web application that is used by thousands of unique clients. White-label products are designed to be customizable on a per-client basis, providing a different look and feel depending on the client's preferences. Each client will want their own unique styling in terms of typography, colors, borders, etc. For the purpose of this exercise, assume customization solely in terms of styling.

The web application you are to design features multiple pages with dynamic content. The manner by which you provide styling to the application is up to you. Referring to the sample screenshots below, explain and/or diagram how you would go about architecting this application. How would you design this system? What are the pros and cons to your approach? Tradeoffs? What design patterns would you use and why? List all assumptions.

Sample Screenshots

LOG IN

MEN WOMEN KIDS SALE

Shop Men / New Arrivals

SORT BY
CATEGORY +
COLOR +
SIZE +
PRICE RANGE +

Remove all filters

Manson Leather Biker Jacket
$1,055.00

Dial Leather Biker Jacket
$630.00

Manta Bomber Jacket
$578.00

Colman Parka Jacket
$630.00

Payne Mac
$520.00

Cause Bomber Jacket
$490.00

Echo Bomber Jacket
$490.00

Camana Cardigan
$208.00

Ettrick Cardigan

Leven Hoody

Iseut Hoody

Arkan Crew Sweat

Log in

MEN WOMEN KIDS SALE

Shop Men / New Arrivals

SORT BY
CATEGORY +
COLOR +
SIZE +
PRICE RANGE +

Remove all filters

Manson Leather Biker Jacket
$1,055.00

Dial Leather Biker Jacket
$630.00

Manta Bomber Jacket
$578.00

Colman Parka Jacket
$630.00

Payne Mac
$520.00

Cause Bomber Jacket
$490.00

Echo Bomber Jacket
$490.00

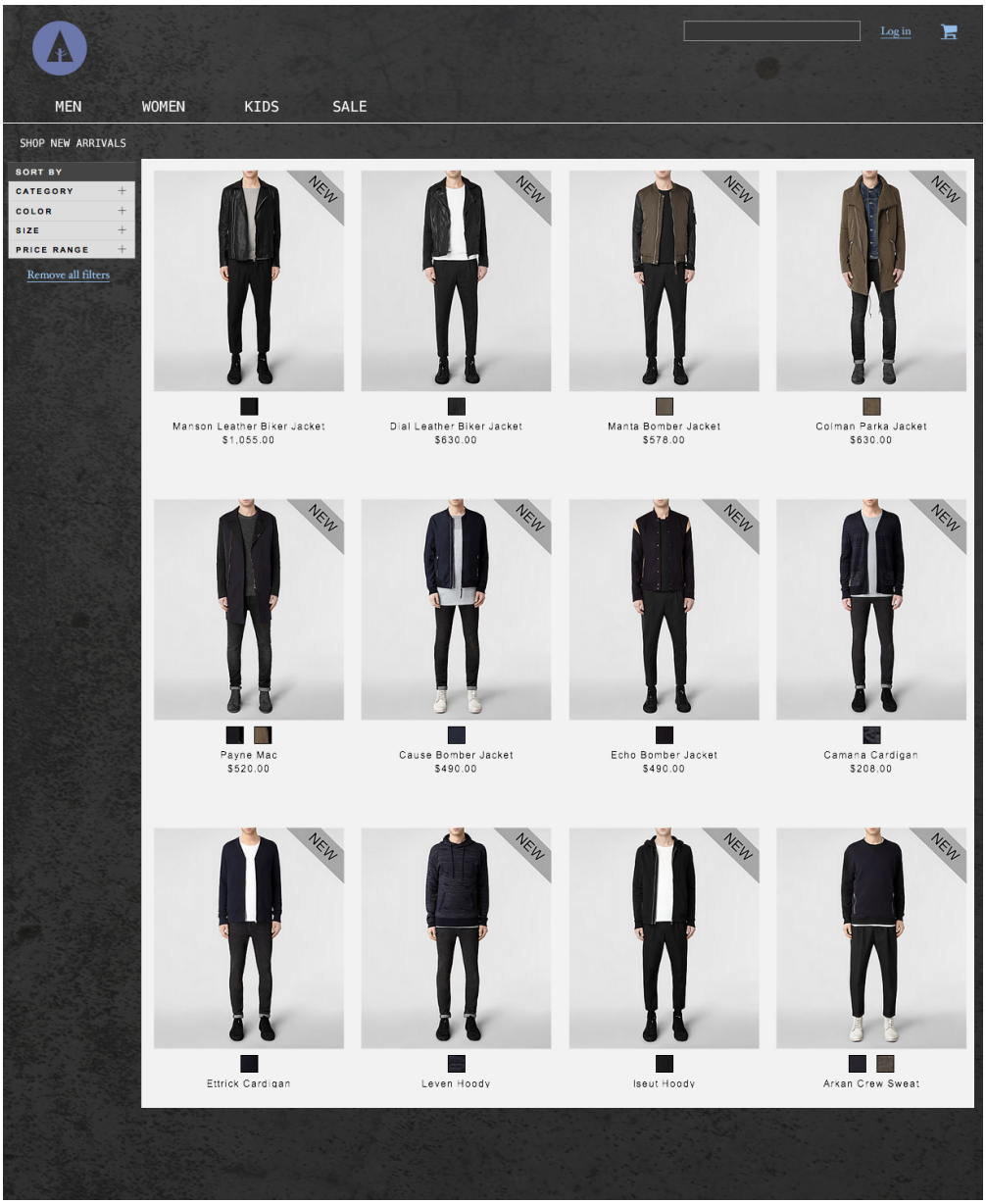
Camana Cardigan
$208.00

Ettrick Cardigan

Leven Hoody

Iseut Hoody

Arkan Crew Sweat


Log in
MEN
WOMEN
KIDS
SALE
SHOP NEW ARRIVALS
SORT BY
CATEGORY
COLOR
SIZE
PRICE RANGE
Remove all filters
NEW
Manson Leather Biker Jacket
$1,055.00
Dial Leather Biker Jacket
$630.00
Manta Bomber Jacket
$578.00
Colman Parka Jacket
$630.00
Payne Mac
$520.00
Cause Bomber Jacket
$490.00
Echo Bomber Jacket
$490.00
Camana Cardigan
$208.00
Ettrick Cardigan
Leven Hoody
Iseut Hoody
Arkan Crew Sweat