1 HTTP (2 points)

a) (1 point) Ligia, qui a suivi avec succès le cours LOG4420, se trouve dans une fête familiale. Rigoberta, sa cousine, est suprise qu'un commerce ait pu savoir quels sites elle a visités sur le web. Ligia lui explique alors le principe. Elle explique d'abord que le commerce en question ne peut suivre la trace de Rigoberta que sur les sites qui affichent une image ou tout autre élément venant du serveur de ce commerce. Complétez l'explication.

Lorsqu'on récupère l'image sur le site du commerce, un cookie est ajouté. Chaque fois qu'une image sera récupérée sur le même site, le cookie sera ajouté à la requête. Il suffit alors d'extraire l'information associée à l'en-tête Referer.

b) (1 point) Le navigateur doit faire une revalidation d'une page en cache, par le biais d'un GET conditionnel. Expliquez les deux approches possibles pour effectuer un GET conditionnel.

Une approche consiste à utiliser l'en-tête `If-modified-since`. Le serveur envoie la ressource demandée si elle a été modifiée depuis la date associée à cet en-tête. La seconde utilise l'en-tête `If-none-match`. La ressource demandée a une étiquette unique et elle est envoyée seulement si cette étiquette est différente de celle fournie avec l'en-tête.

2 HTML et CSS (3 points)

Pour chacun des énoncés suivants, dites s'il est vrai ou faux. Attention : deux mauvaises réponses annulent une bonne réponse (0,5 point pour chaque énoncé).

a) Un des principaux avantages de HTML5 est qu'il a rendu déclaratif, par le biais de nouvelles balises et de nouveaux attributs, ce qui auparavant devait être programmé en Javascript. Vrai

b) Un élément encodé par la balise `<h1>` est toujours affiché de la même manière, peu importe sa position dans un document HTML. Faux

c) La petite icône qui apparaît sur l'onglet, lorsqu'un document HTML est chargé dans le navigateur, est spécifiée par une balise `<img>` dans le document HTML. Faux

d) Soit un élément d'un document HTML pour lequel le navigateur doit déterminer le rendu. Lorsqu'il identifie les directives CSS qui peuvent s'appliquer, il se retrouve avec les deux directives suivantes parmi lesquelles il doit choisir (les deux directives proviennent du site web dont le document est originaire) :

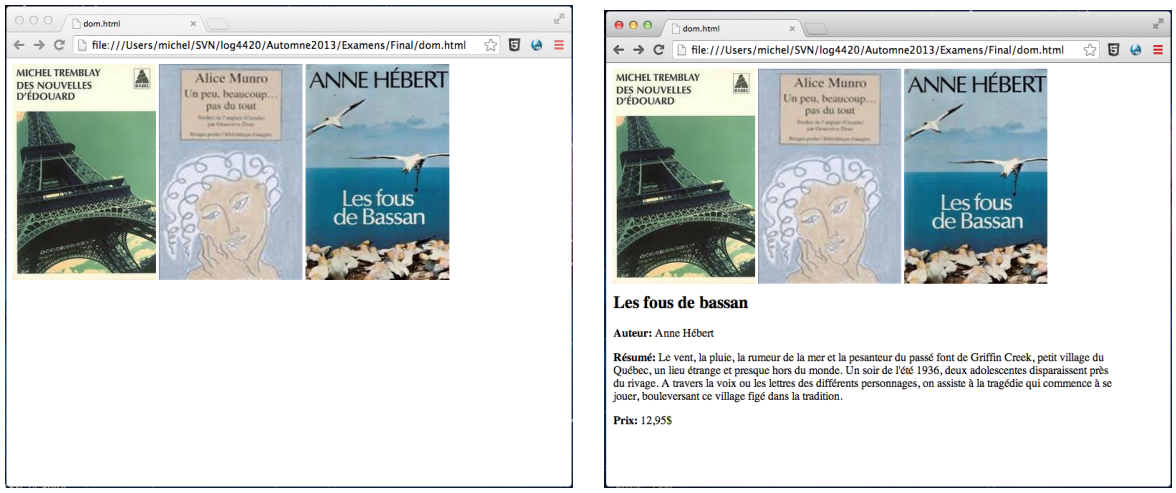
```
div .item .highlight { color: red }
div .item > p { color: blue }
```

L'élément sera affiché en rouge. Vrai

e) Avec CSS, on peut ajouter du contenu à la structure d'un document HTML. Vrai, en utilisant la pseudo-classe `::before` ou `::after`

f) Avec CSS, le positionnement absolu d'une boîte est toujours relatif à la première boîte englobante qui est elle-même positionnée, ou la fenêtre du navigateur si une telle boîte n'existe pas. Vrai

3 DOM et Javascript (4 points)



(a) (b)

FIGURE 1 – Page web dynamique

La figure 1 illustre une page dynamique. En (a), on a l'affichage initial lorsque la page est chargée dans le navigateur. Trois images de livres sont affichées. Lorsqu'on clique sur une image, une description du livre qu'elle représente apparaît en dessous des images. S'il y avait déjà une description auparavant, celle-ci est remplacée par la nouvelle. Écrivez les codes CSS et Javascript (qui se retrouveront dans les fichiers *exemple.css* et *exemple.js*, respectivement) nécessaires pour obtenir ce comportement, sachant que le code source HTML est le suivant (vous ne pouvez pas modifier ce code) :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="exemple.css" />
  </head>
  <body>
    <div class="illustrations">
      <div class="item" id="1">
        <img src="tremblay.jpeg" width="200" height="300"/>
        <article>
          <h1>Des nouvelles d'Édouard</h1>
          <p><strong>Auteur:</strong> Michel Tremblay </p>
          <p><strong>Résumé:</strong> La grande Duchesse de Langeais... </p>
          <p><strong>Prix:</strong> 13,95$ </p>
        </article>
      </div>
      <div class="item" id="2">
        <img src="munro.jpeg" width="200" height="300"/>
        <article>
          <h1>Un peu, beaucoup... pas du tout</h1>
          <p><strong>Auteur:</strong> Alice Munro </p>
          <p><strong>Résumé:</strong> Neuf histoires de femmes...  </p>
          <p><strong>Prix:</strong> 23,45$ </p>
        </article>
```

```
      </div>
      <div class="item" id="3">
        <img src="hebert.jpeg" width="200" height="300"/>
        <article>
          <h1>Les fous de bassan</h1>
          <p><strong>Auteur:</strong> Anne Hébert </p>
          <p><strong>Résumé:</strong> Le vent, la pluie, la rumeur de la me...  </p>
          <p><strong>Prix:</strong> 12,95$ </p>
        </article>
      </div>
    </div>
  </body>
  <script src="http://code.jquery.com/jquery-1.10.1.min.js"></script>
  <script src="exemple.js"></script>
</html>
```

CSS :

```
.item {
    display : inline ;
}
article {
    position : fixed ;
    left : 10px ;
    top : 300px ;
    width : 700px ;
}
```

Javascript :

```
$('article').hide() ;
$('.item > img').click(function() {
    $('article').hide() ;
    $(this).parent().children('article').show()}) ;
```

4 Services web (2 points)

a) (1 point) Soit un service web, comme celui qui a été implémenté dans votre travail pratique, qui retourne une page de l'histoire et les pages suivantes accessibles selon la décision du joueur. Voici un exemple de réponse JSON retournée par ce service :

```
{  "no": 234,
   "content": [ { "type": "paragraph",
                  "text": "Vous plongez votre arme dans l'œil monstrueux..." },
                { "type": "image",
                  "url": "monster_luminous_jelly.png" },
                { "type": "paragraph",
                  "text": "Avant que vous puissiez vous relever..." } ],

   "decision": [ { "text": "Si vous maîtrisez la <i>Discipline Magnakaï...",
                   "next": 190 },
                 { "text": "Si vous ne maîtrisez pas cette discipline...",
                   "next": 121 } ] }
```

Indiquez à quel niveau correspond ce service web, selon la classification de Richardson, et justifiez votre réponse.

Niveau 3. La réponse du serveur indique les liens possibles pour la suite de la navigation.

b) (1 point) Expliquez ce qui distingue les méthodes POST, PUT et PATCH pour leur utilisation dans une requête HTTP à un service web.

POST : ajout d'un élément à une collection. PUT : création d'une nouvelle collection ou remplacement d'un item dans une collection. PATCH : modification d'un item déjà extistant.

5 Programmation du côté serveur (4 points)

a) (1 point) Dans une plateforme de développement comme Yesod, on utilise des gabarits (templates) pour les données HTML, CSS et Javascript qui composeront le document final. Ces gabarits sont combinés par le biais de widgets. Expliquez pourquoi il faut faire appel à des widgets pour obtenir le résultat final.

Le HTML a une structure arborescente. Le templates représentent du contenu qui pourrait devoir être dispersé à plusieurs endroits dans cet arbre.

b) (1 point) Expliquez ce qu'est une route et son rôle sur le serveur.

Une route correspond au chemin spécifié dans l'URL. Elle sert à aiguiller la requête vers le contrôleur adéquat pour la traiter.

c) (1 point) Indiquez deux avantages importants du traitement des formulaires dans une plateforme comme Yesod.

En un seul endroit dans le code, (1) on associe la génération et le (2) traitement du formulaire, tout le code nécessaire étant géré par la plateforme.

d) (1 point) Expliquez comment les sessions sont représentées en Yesod et comment on les manipule.

Les sessions sont représentées par un ensemble de paires attribut/valeur. On a les fonctions `setSession`, `getSessin` et `deleteSession` pour ajouter, obtenir la valeur et retirer une telle paire, respectivement.

6 Plateforme MVC sur le client (3 points)

Soit l'application web illustrée à la figure 2, qui permet essentiellement de gérer une liste de tâches. En (a), on a l'affichage au moment où la page est téléchargée par le navigateur. On remarque qu'une des tâches est déjà cochée et présentée en gris plutôt qu'en noir. En (b), nous avons l'état obtenu après avoir ajouté une tâche dans la boîte de texte et cliqué sur le bouton *ajouter* (à noter que le nombre total de tâches et le nombre de tâches à faire ont été mis à jour). En (c), la deuxième tâche a été sélectionnée et apparaît donc elle aussi en gris. Finalement, en (d), on a l'état après avoir cliqué sur le lien *archive* : les tâche sélectionnées ont été retirées de la liste.

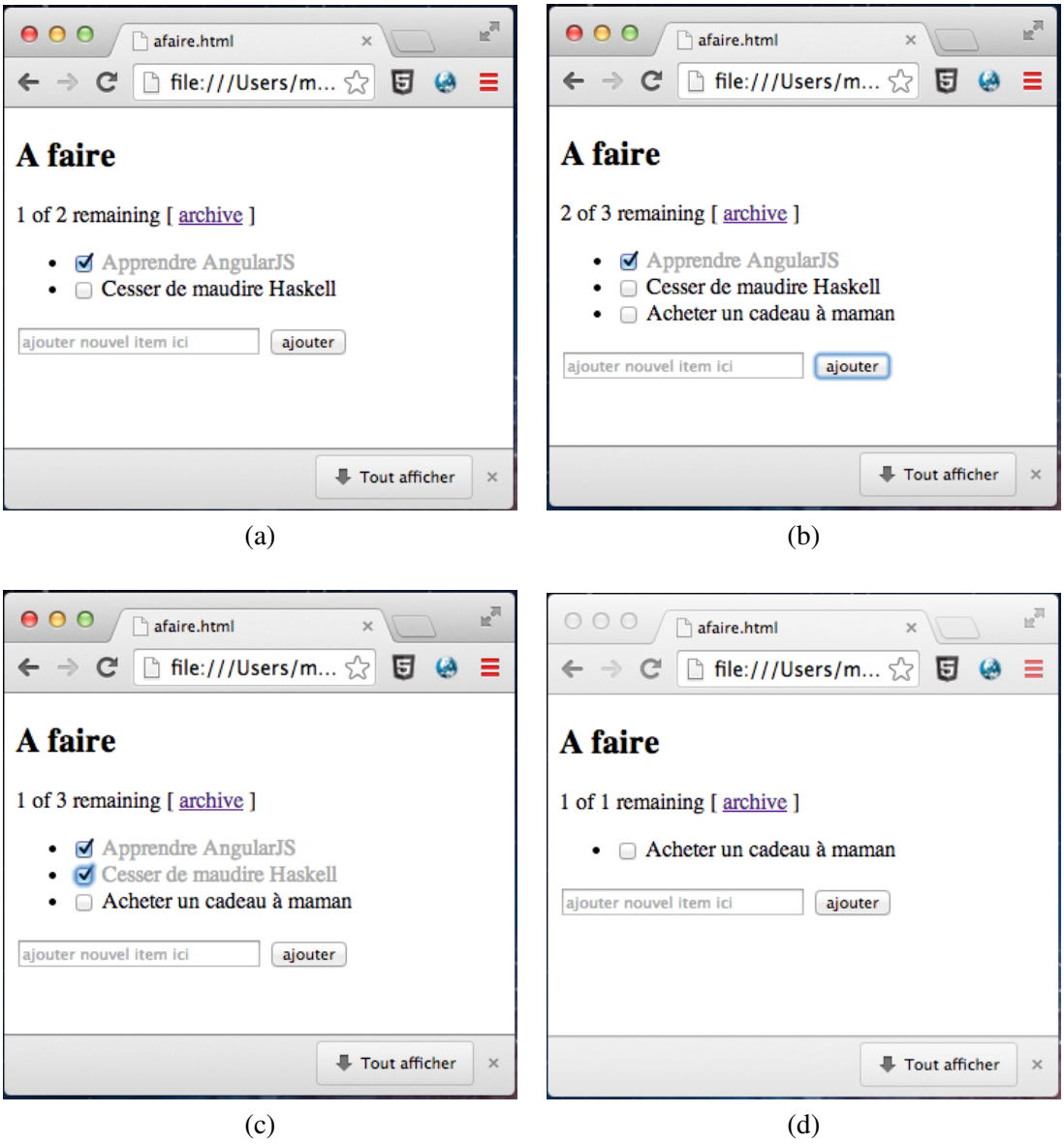


FIGURE 2 – Application web

Pour gérer cette application, on utilise AngularsJS. Le code de la page téléchargée est le suivant :

```
<!DOCTYPE html>
<html>
  <head>
    <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.0.8/angular.min.js"></script>
    <style>
     .done-true {
        color: #aaa;
     }
    </style>
  </head>
  <body>

    <div ng-app>
      <h2>A faire</h2>
      <div ng-controller="TodoCtrl">
        <span>{{remaining()}} of {{todos.length}} remaining</span>
        [ <a href="" ng-click="archive()">archive</a> ]
        <ul>
           <li ng-repeat="todo in todos">
           <input type="checkbox" ng-model="todo.done">
           <span class="done-{{todo.done}}">{{todo.text}}</span>
           </li>
        </ul>
        <form ng-submit="addTodo()">
           <input type="text" ng-model="todoText"  size="30 placeholder="ajouter nouvel item ici">
           <input type="submit" value="ajouter">
        </form>
      </div>
    </div>
  </body>
</html>
```

Écrivez le code Javascript nécessaire pour faire fonctionner cette application. Supposez que tout se passe localement.

```
function TodoCtrl($scope) {
    $scope.todos = [
        {text :'Apprendre AngularJS', done :true},
        {text :'Cesser de maudire Haskell', done :false}] ;

    $scope.addTodo = function() {
        $scope.todos.push({text :$scope.todoText, done :false}) ;
        $scope.todoText = '' ;
        } ;

    $scope.remaining = function() {
        var count = 0 ;
        for (i = 0 ; i < $scope.todos.length ; i++) {
            if ( !$scope.todos[i].done) {
                count += 1
            }
        }
        return count ;
```

```
    } ;

    $scope.archive = function() {
        var oldTodos = $scope.todos ;
        $scope.todos = [] ;
        for (i = 0 ; i < oldTodos.length ; i++) {
            if ( !oldTodos[i].done) {
                $scope.todos.push(oldTodos[i]) ;
            }
        }
    } ;
} ;
```

7 RDF et web sémantique (2 points)

a) (1 point) Représentez la situation suivante en RDF, de manière concise, en utilisant la notation Turtle :

— L'article A1 a été écrit par Michel Gagnon.
— Les articles A2 et A3 ont été écrits par le même auteur.
— L'article A3 cite quelqu'un qui critique l'article A1.

Supposez que les URI pour les articles sont `doc:A1`, `doc:A2` et `doc:A3`. Utilisez les propriétés `dc:author` et `foaf:name`.

```
@prefix doc: <www.exemple.org/doc/> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

doc:A1
      a :Article;
      dc:author [ foaf:name "Michel Gagnon" ] .
doc:A2 a :Article;
      dc:author _:n1;
doc:A3 a :Article;
      dc:author _:n1;
      :cite [ a foaf:Person;
            :critique doc:A1 ] .
```

b) (1 point) Soit la base de données RDF suivante :

```
@prefix voc: <http://exemple.org/Voc/> .
@prefix poly: <http://www.polymtl.ca/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

poly:log4420 poly:professeur poly:MichelGagnon ;
         voc:inscrit  [ foaf:name  "Victor Hugo" ] ;
         voc:inscrit  [ foaf:name  "Papa Noël" ] ;
         voc:inscrit  [ foaf:name  "Roger Lemelin" ] ;
         voc:inscrit  [ foaf:name  "Anne Hébert" ] .

poly:inf6410 poly:professeur poly:MichelGagnon ;
         voc:inscrit  [ foaf:name  "Papa Noël" ] ;
         voc:inscrit  [ foaf:name  "Michel Tremblay" ] ;
         voc:inscrit  [ foaf:name  "Louis Caron" ] .

poly:log3000 poly:professeur poly:PierreRobillard ;
         voc:inscrit  [ foaf:name  "Papa Noël" ] ;
         voc:inscrit  [ foaf:name  "André Langevin" ] ;
         voc:inscrit  [ foaf:name  "Gaëtan Soucy" ] .
```

Écrivez la requête SPARQL qui permet d'extraire les noms de tous les étudiants qui ont suivi au moins un cours avec Michel Gagnon. Vous devez éviter les réponses redondantes.

```
SELECT DISTINCT ?nom WHERE {
    [] poly :professeur poly :MichelGagnon ;
    voc :inscrit [ foaf :name ?nom ]
}
```

8 Question bonus (0,5 point)

Répondez à une seule des questions suivantes :

a) Quel est le mot de passe actuel du professeur de LOG4420 ?

b) Donnez le nom du musicien catalan qui s'est fait connaître pour ses interprétations d'oeuvres composées pour viole de gambe, notamment celles de Marin Marais.

Jordi Savall