# Bookmarks HTML5 Application – Sample JS Apps Exam

You are assigned to design and implement a **bookmarks management Web front-end application** using HTML5, JavaScript, jQuery, AJAX, REST and JSON with cloud-based backend. The app keeps users and their bookmarks. Users can register, login, view their bookmarks, add and delete bookmarks and logout. The app should be implemented as client-side Web application in JavaScript with server-side REST services called by AJAX and returning JSON objects.

## Create Bookmarks REST Services

Register at Parse.com and create an application to keep your data in the cloud. Create a class **Bookmark(title, url)** to hold bookmarks. Parse.com will automatically create the following REST services to access your data:

* **User Registration (Sign Up)**
  + Endpoint: <https://api.parse.com/1/users>, Method: POST
  + Request body (JSON): {"username":"*user*", "password":"*pass*"}
  + Returns (JSON): {…, "sessionToken":"*session\_token*"}
* **User Login**
  + Endpoint: <https://api.parse.com/1/login>, Method: GET
  + URL parameters: username=*user*, password=*pass*
  + Returns (JSON): {…, "sessionToken":"*session\_token*"}
* **List All Bookmarks**
  + Endpoint: <https://api.parse.com/1/classes/Bookmark>, Method: GET
  + Returns (JSON): {"results":[{"title":"…","url":"…",…}, …]}
* **Add Bookmark**
  + Endpoint: <https://api.parse.com/1/classes/Bookmark>, Method: POST
  + Request body (JSON): {"title":"*title*", "url":"*url*", "ACL":{…}}
  + Returns (JSON): {"createdAt":"…", "objectId":"…"}
* **Delete Bookmark**
  + Endpoint: [https://api.parse.com/1/classes/Bookmark/*bookmark\_id*](https://api.parse.com/1/classes/Bookmark/bookmark_id), Method: DELETE
  + Returns (JSON): {}

All Parse.com REST services require the following **HTTP request headers**:

* X-Parse-Application-Id: *your\_parse\_app\_id*
* X-Parse-REST-API-Key: *your\_parse\_rest\_api\_key*

Notes about **users and authentication**:

* After register / login, pass the session token as HTTP request header to **authenticate the request**:
  + X-Parse-Session-Token: *session\_token\_returned\_by\_login*
* When creating new objects, pass the following **ACL** to restrict the access to the user created the object:
  + "ACL":{"*user\_оbjectId*":{"write":true,"read":true}}
  + Thus the object will only be visible and accessible by the specified user and invisible for all others

5 score

## Slice and Dice the Bookmarks Web Design

Slice and dice the Web design of the Bookmarks application into HTML5 + CSS3. The design is given as PowerPoint document (see [Bookmarks.pptx](file:///C:\Software-University\Courses\JavaScript-Applications\November-2014\JS-Apps-Exam-Problems\Bookmarks.pptx)) and will be used to implement the application functionality. Pixel-perfect layout is and responsive design is not required. Use fixed-width layout. You do not need to match sizes / fonts / colors.

20 score

## Bookmarks Client-Side Web Application

Design and implement a client-side web app for managing the bookmarks with the following functionality:

* **Register user** – by username and password the app should be able to register new user in the system. After a successful registration, a green notification message should be displayed and the login screen should be displayed. In case of error, an appropriate error message should be displayed and the user should be able to try to register again.

10 score

* **Login user** – by username and password the app should be able to login an existing user. After successful login, the bookmarks screen should be displayed to view / manage his bookmarks. In case of error, an error message should be displayed and the user should be able to try to login again.

10 score

* **Display user's bookmarks** – after a successful login the system should display all bookmarks created by the current user. The bookmarks should be listed in several columns with fixed width. Note that some bookmark titles and URLs can be long so your bookmark box should be able to resize properly. In case of error (e.g. Internet connection lost), an error message should be displayed.

Ensure you handle property the HTML special characters, e.g. the bookmark title can be "*Working with the <div> and <span> tags in HTML*".

20 score

* **Add new bookmark** – successfully logged in users should be able to create new bookmarks by entering title and URL and clicking the [Add] button. The "New Bookmark" box should be shown after the last displayed bookmark. After successful bookmark creation, a green notification message should be displayed and the last added bookmark should be displayed first. In case of error, an appropriate error message should be displayed and the user should be able to try to add the bookmark again.

10 score

* **Delete a bookmark** – successfully logged in users should be able to delete their bookmarks by clicking at the bookmark's [x] button. Deleting needs a confirmation from the user. After successful bookmark deletion, a green notification message should be displayed and the bookmark should be removed. In case of error, an appropriate error message should be displayed.

10 score

* **Logout** – successfully logged in user should be able to logout from the app. After a successful logout, a green notification message should be displayed and the login screen should be shown.

5 score

* **Notifications** – the application should notify the users about the result of their actions. In case of success a green notification message should be shown, which disappears automatically after 5 seconds or manually when the user clicks its [x] button. In case of error, a red notification message should be shown which disappears when the user clicks its [x] button.

10 score