Course:	**Web Programming**	**Course Code:**	**CS-440**
Program:	**BS (Computer Science)**	**Semester:**	**Spring 2018**
Duration:	**180 Minutes**	**Total Marks:**	**80**
Paper Date:	**30-May-18**	**Weight**	**40%**
Section:	**A & B**	**Page(s):**	**2**
Exam:	**Final**	**Reg. No.**	

Instruction/Notes: This exam is open books and open notes.

Consider the case of a sports related web application (such as CricInfo) targeting Cricket followers. The application provides live scorecard of a Cricket match happening anywhere on the globe to its interested users. A match is played between two teams and identified with: names of the two teams combined with the date, the match is played on, in unordered form. For instance, (Pakistan vs England, 24-May-2018) and (England vs Pakistan, 24-May-2018) refers to the same match.

Question 1 **30 points**

The application relies upon a webservice to propagate the live scorecard. The Webservice supports a function to query match information, comprising the following:

- **Type:** The type of match can be one of three: Test, One-day, or T20.
- **State:** The state can be either **live** indicating that the match is being played, **break** indicating a scheduled break (e.g. lunch, tea, innings, etc.), **halted** indicating the match is temporarily stopped for unexpected reasons (e.g. rain, etc.), **finished** indicating the match is over. A textual **description** may also accompany the state for clarity (e.g. "match is delayed due to rain", etc.).
- **Result:** The final result of the match, either with the information of winners and losers, or that it ends in draw. Its applicable only if the state is **finished**
- **Score:** Prior score information for each team (for all the innings already played in the match) as well as the current score (runs on the scoreboard, wickets fallen, and overs and balls bowled) while also indicating the current batting and bowling sides.

Design and implement the above function in a REST based webservice, and produce:

1. URI structure to invoke the webservice function and details of the input parameters (5 points)
2. **XML** based response message structure capturing complete information (10 points)

3. Server-side code (in any technology of choice) for implementation (10 points)
4. Support for cross-domain calls using **JSONP** (5 points)

Question 2 **20 points**

Develop a Javascript widget that can make **cross-domain AJAX** calls to the webservice developed in Question-1 and show the live score information (as shown in figure below) by polling (repeatedly query) the webservice with a configurable time interval that may default to 30 seconds. You need to provide your own implementation for cross-domain AJAX and cannot rely upon any framework.

Live, Test, England vs Pakistan May 24-28 2018	
England	184 & 242
Pakistan *	363 & 62/1 (12.3 overs)

Question 3 **10 points**

Suppose, in order to improve the performance, you are asked to redesign the implementation of the live score functionality (as implemented in questions 1&2) using **Web Sockets**. Highlight what additions / changes may be required to the existing code? Complete implementation is not required.

Question 4 **20 points**

Considering the afore-mentioned scenario, briefly answer:

1. Can you use idea of Cache for improving the performance in the above case? Justify your decision, with reference to information you may cache. (5 points)
2. How to make your implementation scalable? (5 points)
3. Is your implementation vulnerable to SQL Injection or XSS attack(s)? How? (5 points)
4. Is there a need for using SSL in this scenario? Analyze its pros and cons. (5 points)