USN | | | | | | | | | | |

13MCA43

Fourth Semester MCA Degree Examination, June/July 2017

Advanced Web Programming

Time: 3 hrs. Max. Marks:100

Note: ***Answer any FIVE full questions.***

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

1 a. Explain Perl arrays with creation methods. How Perl arrays differ from other arrays? **(06 Marks)**
b. Explain with examples any two string functions in Perl. **(04 Marks)**
c. Explain the basics of pattern matching, substitutions and transliterate in Perl. **(10 Marks)**

2 a. List and explain some of the common CGI.pm functions. **(06 Marks)**
b. Explain steps in creation of dynamic html pages using CGI. **(08 Marks)**
c. Explain methods in adding robustness to CGI scripts. **(06 Marks)**

3 a. Write a Perl program to insert name and age information entered by the user in to a table created using MySQL and to display the current content of this table. **(10 Marks)**
b. What is the benefit of Perl DBI? Explain Perl DBI methods. **(10 Marks)**

4 a. List and explain syntactic characteristics and uses of PHP. **(10 Marks)**
b. Explain file handling functions in PHP with examples. **(10 Marks)**

5 a. Explain in brief PHP's XML facilities. **(06 Marks)**
b. Explain built-in methods for arrays and lists in Ruby. **(10 Marks)**
c. Explain syntactic container for class in Ruby. **(04 Marks)**

6 a. Write function definition in Ruby for swapping of two numbers. **(05 Marks)**
b. Explain with figure directory structure of rails application. **(08 Marks)**
c. Give an example how dynamic documents are generated in Ruby on Rails. **(07 Marks)**

7 a. What are social networkings? Briefly discuss about floksonomies. **(06 Marks)**
b. Explain SaaS and multiple delivery channels in Web 2.0. **(08 Marks)**
c. Discuss how the rich browser experience is made possible. **(06 Marks)**

8 a. What is JSON? Explain the different types of literals used in JSON with examples. **(10 Marks)**
b. Write short notes on SOAP, WSDL and REST services. **(10 Marks)**

* * * * *

1a	Explain Perl arrays with creation methods. How does a Perl array differ from other array.
	
	An array is a variable that stores an ordered list of scalar values. Array variables are preceded by an "at" (@) sign. To refer to a single element of an array, you will use the dollar sign ($) with the variable name followed by the index of the element in square brack Array variables are prefixed with the @ sign and are populated using either parentheses or the qw operator. For example –
	
	@array = (1,2,3,4);
	
	@array = qw(a b c d);
	
	Print $array[0];
	
	@array = (1..10);
	
	Foreach(@array as $k)
	
	Print $k;
		6

1 b	Explain any two string functions in Perl **chomp** Removes line ending characters from a string or array of strings. **chop** Removes the last character from a string or array of strings. chop STRING **eval** Evaluates perl code, then executes it. eval STRING	
1 c	Explain pattern matching in Perl Three types of pattern matching Match operator $string =~ m/pattern/ Substitute operator $string =~ s/pattern/substitute	

	Trasliterate operator $string =~ tr/[a-z]/[A-Z]/	
2a	List some common CGI.pm functions Start_html() Instead of the normal <html> tags Header() To set the content type to text/html Print table Table tags Print hr Horizontal rule	
2b	Explain steps in creating dynamic pages using CGI There are two methods for sending form data: GET and POST. The main difference between these methods is the way in which the form data is passed to the CGI program. If the GET method is used, the query string is simply appended to the URL of the program when the client issues the request to the server. This query string can then be accessed by using the environment variable QUERY_STRING. Here is a sample GET request by the client, which corresponds to the first form example: GET /cgi-bin/program.pl?user=Larry%20Bird&age=35&pass=testing HTTP/1.0 Accept: www/source Accept: text/html Accept: text/plain User-Agent: Lynx/2.4 libwww/2.14 The GET method has both advantages and disadvantages. The main advantage is that you can access the CGI program with a query without using a form. A query is one method of passing information to a CGI program via the URL. The other method involves sending extra path information to the program. The main advantage to the POST method is that query length can be unlimited-- you don't have to worry about the client or server truncating data. To get data sent by the POST method, the CGI program reads from standard input. There are a number of libraries available in the CPAN(Comprehensive Perl Archive Network) , sites which help with the problem of parsing input from the user. They handle difficult MIME types very effectively . The most popularly used module is the CGI.pm module. When a user submits a web form the contents of the form are extracted by the browser and packaged as a message which is returned to a web server. The browser transmits the data to the server. The server must then pass the data to an appropriate application for further	

processing .
The CGI protocol expects that you are going to be sending some data back to the browser. If there is no provision for this in the script written the web server will handle this task. The apache server sends back an html page which says that there has been an internal server error due to misconfiguration of the web server.

It also writes a message in the error log which says “premature end of script headers”

This means the script did not follow the protocol by correctly formatting the HTTP message in response to data from the web form. The CGI.pm module is used to parse data and to create the page which will be returning to the browser. The module is complete and includes a lot of functionalities

2c Explain adding robustness in PERL

Any CGI script that processes information from users is a security risk.

It is a designed to increase the security by preventing malicious users from executing commands on a host computer.

It highlights specific security risks primarily associated with websites which are attacked with techniques such as SQL injection , buffer overflow etc.

The concept behind taint checking is that any variable that can be modified by an outside user poses a potential security risk.

The unchecked variables supplied by the user can be passed directly to the operating system shell.

To turn on taint checking one has to pass –T flag to Perl when it is invoked. The flag turns on runtime taint checking. It will not cause program to fail to compile due to syntax errors but it will cause program to exit if tainted data is passed to a shell command.

3a Write a perl program to insert name and age information entered by the user into a table created using MySQL and to display the current contents of this table.

Go to Filesystem\var\www\html

Create a file main.html

```
<html>
<body>
<form  action="http://localhost/cgi-bin/page.pl">
<input type="text" name="t1">
<input type="text" name="t2">
<input type="submit" value="submit">
</form>
</body>
</html>
```

Go to Filesystem\var\www\cgi-bin

Create a file page.pl

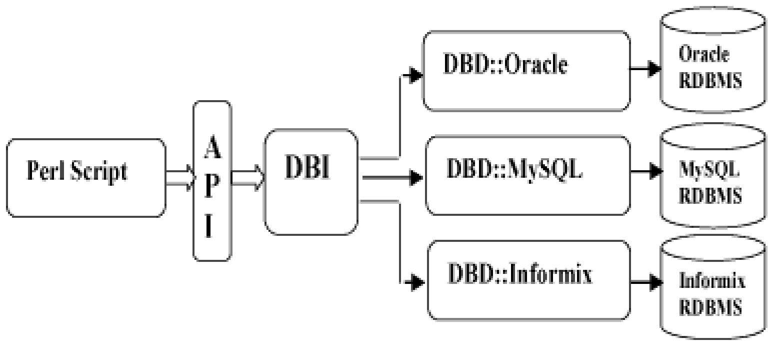
```
#!/usr/bin/perl  -w
use CGI ':standard', '-debug';
print "content-type:text/html","\n\n" ;
use DBI;
$dbh=DBI->connect("DBI:mysql:student1","root","");
$name=param("t1");
$age=param("t2");
$sth=$dbh->prepare("insert into stud1 values('$name','$age')");
$sth->execute();
$dh=$dbh->prepare("select  * from stud1");
$dh->execute();
while(($name,$age) = $dh->fetchrow())
{
        print "$name";
        print "$age";
}
$dh->finish();
$dbh->disconnect();
```

3b Explain Perl DBI and its methods?

DBI stands for **Database Independent Interface** for Perl which means DBI provides an abstraction layer between the Perl code and the underlying database, allowing you to switch database implementations really easily. The DBI is a database access module for the Perl programming language. It provides a set of methods, variables, and conventions that provide a consistent database interface, independent of the actual database being used.

Architecture of a DBI Application

DBI is independent of any database available in backend. You can use DBI whether you are working with Oracle, MySQL or Informix etc. This is clear from the following architure diagram.



```
my $driver = "mysql";
my $database = "TESTDB";
my $dsn = "DBI:$driver:database=$database";
my $userid = “root”;
my $password = “ “;
my $dbh = DBI->connect($dsn, $userid, $password )
```

If a connection is established with the datasource then a Database Handle is returned and saved into $dbh for further use otherwise $dbh is set to *undef* value

Insert

	Prearing SQL statement with INSERT statement. This will be done using **prepare()** API. Executing SQL query to select all the results from the database. This will be done using **execute()**API. Releasing Stattement handle. This will be done using **finish()** API READ Operation READ Operation on any database means to fetch some useful information from the database ie one or more records from one or more tables. So once our database connection is established, we are ready to make a query into this database. Following is the procedure to query all the records having AGE greater than 20. This will take four steps Preparing SQL SELECT query based on required conditions. This will be done using **prepare()**API. Executing SQL query to select all the results from the database. This will be done using **execute()**API. Fetching all the results one by one and printing those results.This will be done using **fetchrow()** API. Releasing Stattement handle. This will be done using **finish()** API Disconnecting Database To disconnect Database connection, use **disconnect** API as follows: `$rc = $dbh->disconnect();` Similarly for update and delete operation	
4a	List out syntactic characteristics and uses of PHP. **PHP** is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994,[4] the PHP reference implementation is now produced by The PHP Development Team.[5] PHP originally stood for *Personal Home Page*,[4] but it now stands for the recursive acronym *PHP: Hypertext Preprocessor*.[6] PHP code may be embedded into HTML or HTML5 markup, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server software combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.[7] The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.[8] The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a *de facto* standard. Since 2014 work has gone on to create a formal PHP specification	
4b	Explain file handling in PHP a. Opening files	

To open files we use fopen() function.
Fopen(name of the file, mode)

Modes are
r : read mode
w: write mode
a: append mode

```
<?php
$fp = fopen("file.txt","r");
Echo fread($fp);
Fclose($fp);
?>
```

Reading contents from a file

Fread , file_get_contents , fgets etc are similar functions for reading contents of a file.
The contents of the file file.txt will be displayed when the program is executed.

Writing to a file
File_put_contents($fp,"contents to be written");

Closing a file
Fclose ($fp);

5a Explain PHP XML facilities

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<student_info>
<student>
<usn>1CR10MCA01</usn>
<name>Ajay</name>
</student>
<student>
<usn>1CR10MCA02</usn>
<name>Akshatha</name>
</student>
<student>
<usn>1CR10MCA58</usn>
<name>Piyush</name>
</student>
<student>
<usn>1CR10MCA59</usn>
<name>Taj</name>
</student>
</student_info>
```

3.php

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
  {
die('Could not connect: ' . mysql_error());
```

```
  }

mysql_select_db("mydb", $con);
 $lib  = simplexml_load_file("3.xml");
//if data is already existing then delete
$i="delete from stu";
$result=mysql_query($i);

foreach($lib as $stu){

$usn= $stu->usn;
$name=$stu->name;
$i="insert into student(usn,name) values('$usn','$name')";
mysql_query($i);
}
$result = mysql_query("SELECT * from student");
//Use the mysql_fetch_array function on the resource returned by the mysql_query
$row = mysql_fetch_array($result);
//Display the results
echo "<table border='1'><tr><th>USN</th><th>Name</th></tr>";

while($row = mysql_fetch_array($result)){

echo "<tr><td>" . $row['usn'] . "</td><td>" . $row['name'] . "</td></tr>";


}
echo "</table>";
?>
```

5b Explain built in methods for arrays in pHP

An array is a data structure that stores one or more similar type of values in a single value. For example if you want to store 100 numbers then instead of defining 100 variables its easy to define an array of 100 length.

There are three different kind of arrays and each array value is accessed using an ID c which is called array index.

- **Numeric array –** An array with a numeric index. Values are stored and accessed in linear fashion.
- **Associative array –** An array with strings as index. This stores element values in association with key values rather than in a strict linear index order.
- **Multidimensional array –** An array containing one or more arrays and values are accessed using multiple indices

Numeric Array

These arrays can store numbers, strings and any object but their index will be represented by numbers. By default array index starts from zero.

Example

Following is the example showing how to create and access numeric arrays.Here we have used **array()** function to create array. This function is explained in function reference.

```
<html>
  <body>

    <?php
      /* First method to create array. */
      $numbers = array( 1, 2, 3, 4, 5);

      foreach( $numbers as $value ) {
        echo "Value is $value <br />";
      }

      /* Second method to create array. */
      $numbers[0] = "one";
      $numbers[1] = "two";
      $numbers[2] = "three";
      $numbers[3] = "four";
      $numbers[4] = "five";

      foreach( $numbers as $value ) {
        echo "Value is $value <br />";
      }
    ?>

  </body>
</html>
```

Associative Arrays

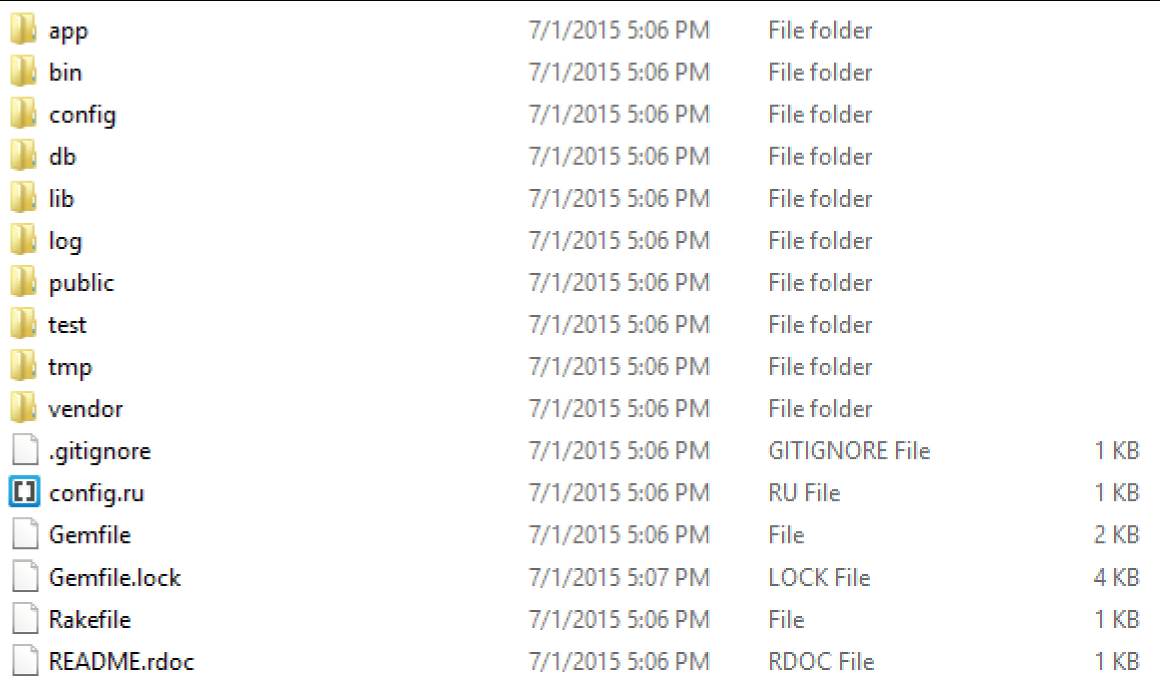
The associative arrays are very similar to numeric arrays in term of functionality but they are different in terms of their index. Associative array will have their index as string so that you can establish a strong association between key and values.

```
<html>
  <body>

    <?php
      /* First method to associate create array. */
      $salaries = array("mohammad" => 2000, "qadir" => 1000, "zara" => 500);

      echo "Salary of mohammad is ". $salaries['mohammad'] . "<br />";
      echo "Salary of qadir is ".  $salaries['qadir']. "<br />";
      echo "Salary of zara is ".  $salaries['zara']. "<br />";

      /* Second method to create array. */
      $salaries['mohammad'] = "high";
      $salaries['qadir'] = "medium";
      $salaries['zara'] = "low";

      echo "Salary of mohammad is ". $salaries['mohammad'] . "<br />";
      echo "Salary of qadir is ".  $salaries['qadir']. "<br />";
      echo "Salary of zara is ".  $salaries['zara']. "<br />";
    ?>
```

5c	Explain container calss in Ruby Containers are objects that hold reference to other objects Commonly used containers String Ordered container of characters Array Like strings but can hold objects of any type Occur naturally in many applications In mp3 player playlist is a container of songs In GUI window is container of widgets (buttons,textbox etc) Do not need to keep track of related objects individually Easy to operate on all the related objects simultaneously Examples Shuffle the songs in a playlist Display all the widgets in a GUI Easy to add, destroy, maintain the objects in container.	

6a	Write function definition in Ruby for swapping#
	#!/usr/bin/Ruby
	&swap;
	
	Sub swap()
	{
	A = gets.to_i;
	B=gets.to_i;
	t.to_i = a;
	a=b;
	b=t;
	puts a;
	puts b;
	}	

6b	Explain figure directory structure of Rails When you use the Rails helper script to create your application, it creates the entire directory structure for the application. Rails knows where to find things it needs within this structure, so you don't have to provide any input. Here is a top-level view of a directory tree created by the helper script at the time of application creation. Except for minor changes between releases, every Rails project will have the same structure, with the same naming conventions. This consistency gives you a tremendous advantage; you can quickly move between Rails projects without relearning the project's organization. When we type **rails new bookstore** It creates a new application with the name bookstore. You will find a directory structure in Windows as follows –	



```
app          7/1/2015 5:06 PM   File folder
bin          7/1/2015 5:06 PM   File folder
config       7/1/2015 5:06 PM   File folder
db           7/1/2015 5:06 PM   File folder
lib          7/1/2015 5:06 PM   File folder
log          7/1/2015 5:06 PM   File folder
public       7/1/2015 5:06 PM   File folder
test         7/1/2015 5:06 PM   File folder
tmp          7/1/2015 5:06 PM   File folder
vendor       7/1/2015 5:06 PM   File folder
.gitignore   7/1/2015 5:06 PM   GITIGNORE File   1 KB
config.ru    7/1/2015 5:06 PM   RU File          1 KB
Gemfile      7/1/2015 5:06 PM   File             2 KB
Gemfile.lock 7/1/2015 5:07 PM   LOCK File        4 KB
Rakefile     7/1/2015 5:06 PM   File             1 KB
README.rdoc  7/1/2015 5:06 PM   RDOC File        1 KB
```

- **app** – It organizes your application components. It's got subdirectories that hold the view (views and helpers), controller (controllers), and the backend business logic (models).

- **app/controllers** – The controllers subdirectory is where Rails looks to find the controller classes. A controller handles a web request from the user.

- **app/helpers** – The helpers subdirectory holds any helper classes used to assist the model, view, and controller classes. This helps to keep the model, view, and controller code small, focused, and uncluttered.

- **app/models** – The models subdirectory holds the classes that model and wrap the data stored in our application's database. In most frameworks, this part of the application can grow pretty messy, tedious, verbose, and error-prone. Rails makes it dead simple!

- **app/view** – The views subdirectory holds the display templates to fill in with data from our application, convert to HTML, and return to the user's browser.

- **app/view/layouts** – Holds the template files for layouts to be used with views. This models the common header/footer method of wrapping views. In your views, define a layout using the <tt>layout:default</tt> and create a file named default.html.erb. Inside default.html.erb, call <% yield %> to render the view using this layout.

- **components** – This directory holds components, tiny self-contained applications that bundle model, view, and controller.

- **config** – This directory contains the small amount of configuration code that your application will need, including your database configuration (in database.yml),

6c	your Rails environment structure (environment.rb), and routing of incoming web requests (routes.rb). You can also tailor the behavior of the three Rails environments for test, development, and deployment with files found in the environments directory. • **db** – Usually, your Rails application will have model objects that access relational database tables. You can manage the relational database with scripts you create and place in this directory. • **doc** – Ruby has a framework, called RubyDoc, that can automatically generate documentation for code you create. You can assist RubyDoc with comments in your code. This directory holds all the RubyDoc-generated Rails and application documentation. • **lib** – You'll put libraries here, unless they explicitly belong elsewhere (such as vendor libraries). • **log** – Error logs go here. Rails creates scripts that help you manage various error logs. You'll find separate logs for the server (server.log) and each Rails environment (development.log, test.log, and production.log). • **public** – Like the public directory for a web server, this directory has web files that don't change, such as JavaScript files (public/javascripts), graphics (public/images), stylesheets (public/stylesheets), and HTML files (public). • **script** – This directory holds scripts to launch and manage the various tools that you'll use with Rails. For example, there are scripts to generate code (generate) and launch the web server (server). • **test** – The tests you write and those that Rails creates for you, all goes here. You'll see a subdirectory for mocks (mocks), unit tests (unit), fixtures (fixtures), and functional tests (functional). • **tmp** – Rails uses this directory to hold temporary files for intermediate processing. • **vendor** – Libraries provided by third-party vendors (such as security libraries or database utilities beyond the basic Rails distribution) go here. Apart from these directories, there will be two files available in bookstore directory. • **README** – This file contains a basic detail about Rail Application and description of the directory structure explained above. **Rakefile** – This file is similar to Unix Makefile, which helps with building, packaging and testing the Rails code. This will be used by rake utility supplied along with the Ruby installation.	
6c	Explain how dynamic documents are generated in ROR A recommended work flow for creating Rails Application is as follows –	

	• Use the rails command to create the basic skeleton of the application. • Create a database on the PostgreSQL/sqlite server to hold your data. • Configure the application to know where your database is located and the login credentials for it. • Create Rails Active Records (Models), because they are the business objects you'll be working with in your controllers. • Generate Migrations that simplify the creating and maintaining of database tables and columns. • Write Controller Code to put a life in your application. Create Views to present your data through User Interface	
7a	Explain social networking The social network is a theoretical construct useful in the social sciences to study relationships between individuals, groups, organizations, or even entire societies (social units, see differentiation). A **social network** is a social structure made up of a set of social actors (such as individuals or organizations), sets of dyadic ties, and other social interactions between actors. The social network perspective provides a set of methods for analyzing the structure of whole social entities as well as a variety of theories explaining the patterns observed in these structures. Social networks and the analysis of them is an inherently interdisciplinary academic field which emerged from social psychology ,sociology, statistics, and graph theory. The term is used to describe a social structure determined by such interactions. The ties through which any given social unit connects represent the convergence of the various social contacts of that unit. This theoretical approach is, necessarily, relational. An axiom of the social network approach to understanding social interaction is that social phenomena should be primarily conceived and investigated through the properties of relations between and within units, instead of the properties of these units themselves. Thus, one common criticism of social network theory is that individual agency is often ignored. A **social networking service** (also **social networking site**, **SNS** or **social media**) is a platform to build social networks or social relations among people who share similar interests, activities, backgrounds or real-life connections. The variety of stand-alone and built-in social networking services currently available in the online space introduces challenges of definition.	
7b	Explain SAAS and multiple delivery channels Software as a Service (SaaS) is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network, typically the Internet. SaaS is becoming an increasingly prevalent delivery model as	

underlying technologies that support Web services and service-oriented architecture (SOA) mature and new developmental approaches, such as Ajax, become popular. Meanwhile, broadband service has become increasingly available to support user access from more areas around the world.
SaaS is typically accessed by users using a thin client via a web browser. SaaS has become a common delivery model for many business applications, including office and messaging software etc

SaaS applications are often updated more frequently than traditional software,[18] in many cases on a weekly or monthly basis. This is enabled by several factors:

- The application is hosted centrally, so an update is decided and executed by the provider, not by customers.
- The application only has a single configuration, making development testing faster.
- The application vendor does not have to expend resources updating and maintaining backdated versions of the software, because there is only a single version.[19]
- The application vendor has access to all customer data, expediting design and regression testing.
- The solution provider has access to user behavior within the application

Key Characteristics of SAAS

- Network based access to and management of commercially available software activities that are managed from central locations rather than at user's side.
- Enable users to access applications remotely via the web application delivery that are one to many model
- Centralised feature updating , which eliminates the need for downloadable patch and upgrade.
- These applications are generally priced on a per user basis and often with additional fees for extra bandwidth and storage.

Advantages of SAAS

- Easy to Use : They do not require web browsers to run
- **Lower costs :** SaaS has a differential regarding costs since it usually resides in a shared or multitenant environment where the hardware and software license costs are low compared with the traditional mode
- **Scalability and integration:** Usually, SaaS solutions reside in cloud environments that are scalable and have integration with other SaaS offerings. Comparing with the traditional model, users do not have to buy another server or software. They only need to enable a new SaaS offering and, in terms of server capacity planning, the SaaS provider will own that.
- **New releases (upgrades):** SaaS providers upgrade the solution and it becomes available for their customers. Costs and effort associated with upgrades and new releases are lower than the traditional model that usually forces the user to buy an upgrade package and install it, or pay for specialized services to get the environment upgraded.
- **Compatible:** Due to the delivery nature of SAAS through the internet, applications are able to run on wide variety of devices
- **Faster changes:** Velocity of changes in SAAS applications are faster . The changes are frequent and on demand. Most services are updated every 2 weeks.

Disadvantages of SAAS

- SAAS software may not be robust, as traditional software applications due to browser limitations.
- Privacy : Since all the user's data reside in the cloud there are security and privacy issues.They are usually prone to hacking .
- If the SAAS provider goes down, a wide range of dependant clients could be affected.

7c Explain rich browser experience is made possible

Rich User Experience

Traditional web are built with HTML and CSS、CGI and had been offered as a static page . On the other hand Web 2.0 uses Ajax（Asynchronous JavaScript + XML) presenting dynamic , rich user experience to users .

For example, Google Provided Google Maps and Google Suggest

8a Explain in detail JSON?

JSON: **J**ava**S**cript **O**bject **N**otation. JSON is a syntax for storing and exchanging data. JSON is an easier-to-use alternative to XML. A common use of JSON is to read data from a web server, and display the data in a web page.

JSON Arrays Literals

JSON arrays are written inside square brackets.

Just like JavaScript, a JSON array can contain multiple objects:

```
{"employees":[
   {"firstName":"John", "lastName":"Doe"},
   {"firstName":"Anna", "lastName":"Smith"},
   {"firstName":"Peter", "lastName":"Jones"}
]}
```

The JSON format is syntactically identical to the code for creating JavaScript objects. Because of this similarity, instead of using a parser (like XML does), a JavaScript program can use standard JavaScript functions to convert JSON data into native JavaScript objects.

For AJAX applications, JSON is faster and easier than XML:

Using XML

- Fetch an XML document
- Use the XML DOM to loop through the document

- Extract values and store in variables

Using JSON

- Fetch a JSON string
- JSON.Parse the JSON string

JSON Syntax Rules

JSON syntax is derived from JavaScript object notation syntax:

- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays

JSON values can be:

- A number (integer or floating point)
- A string (in double quotes)
- A Boolean (true or false)
- An array (in square brackets)
- An object (in curly braces)
- null

JSON format supports the following data types –

Type	Description
Number	double- precision floating-point format in JavaScript
String	double-quoted Unicode with backslash escaping
Boolean	true or false
Array	an ordered sequence of values
Value	it can be a string, a number, true or false, null etc
Object	an unordered collection of key:value pairs
Whitespace	can be used between any pair of tokens

null	empty

String Literals

- It is a sequence of zero or more double quoted Unicode characters with backslash escaping.
- Character is a single character string i.e. a string with length 1.

The table shows string types –

Type	Description
"	double quotation
\	reverse solidus
/	solidus
b	backspace
f	form feed
n	new line
r	carriage return
t	horizontal tab
u	four hexadecimal digits

JSON Mixed Literals

It is possible to mix object and array literals creating an array of objects or an object containing an array.

```
Var cars = [
{
“color”:”red”,
“doors”:2,
“paid”:true
},
{
“color”:”blue”,
“doors”:4,
"paid":false
}
];
```

The array cars has 3 objects, The three objects has properties named color, doors, paid.
Alert(cars[1].doors) will display 2

8b Write short notes on SOAP, REST, WSDL

SOAP

SOAP is an XML-based protocol for exchanging information between computers.

- SOAP is a communication protocol.
- SOAP is for communication between applications.
- SOAP is a format for sending messages.
- SOAP is designed to communicate via Internet.
- SOAP is platform independent.
- SOAP is language independent.
- SOAP is simple and extensible.
- SOAP allows you to get around firewalls.
- SOAP will be developed as a W3C standard.

WSDL

WSDL is an XML-based language for describing web services and how to access them.

- WSDL stands for Web Services Description Language.
- WSDL was developed jointly by Microsoft and IBM.
- WSDL is an XML based protocol for information exchange in decentralized and distributed environments.
- WSDL is the standard format for describing a web service.
- WSDL definition describes how to access a web service and what operations it will perform.
- WSDL is a language for describing how to interface with XML-based services.
- WSDL is an integral part of UDDI, an XML-based worldwide business registry.
- WSDL is the language that UDDI uses.

REST

Representational State Transfer (REST) is an architectural **style** that specifies constraints, such as the uniform interface, that if applied to a web service induce desirable properties, such as performance, scalability, and modifiability, that enable services to work best on the Web.

RESTful Web Services are REST architecture based web services. In REST Architecture everything is a resource. RESTful web services are light weight, highly scalable and maintainable and are very commonly used to create APIs for web based applications.

In REST architecture, a REST Server simply provides access to resources and REST client accesses and presents the resources. Here each resource is identified by URIs/ global IDs. REST uses various representations to represent a resource like text, JSON and XML. Now a days JSON is the most popular format being used in web services.

HTTP Methods

Following well known HTTP methods are commonly used in REST based architecture.

- **GET** - Provides a read only access to a resource.
- **PUT** - Used to create a new resource.
- **DELETE** - Used to remove a resource.
- **POST** - Used to update a existing resource or create a new resource.

Properties of a REST Application

The REST style is characterized by the following properties:

- Communication takes place on call. The Client is active and requests a representation from the passive server and/or modifies a resource.
- A resource can be addressed by an unique URI.
- The client can request the representation of a resource in form of a document.
- Representations can refer to further resources.
- The server does not monitor the status of its clients. Each query to the server must contain all information that are necessary for interpreting itself.
- Caching is supported. The server can mark its answers as cacheable or not cacheable.