BCA (4th Semester)

CS4006 : Open Source Web based Programming

Practical Workbook

SET-A

Course Objective: To provide knowledge of web programming to design and develop open source web based application.

Course Outcomes:
CO1: Describe the basic concepts of programming with open source technology.
CO2: Validate user input, manage data flow amongst web pages and apply state management techniques.
CO3: Understand the usage of data management using file and database as well as extends its security.
CO4: Develop and implement an application that manages asynchronous data.
CO5: Design, develop and use applications by creating and using remote service.

Programme Outcomes:
PO1: Ability to understand the concepts of key areas in computer science.
PO2: Ability to design and develop system, component or process as well as test and maintain it so as to provide promising solutions to industry and society.
PO3: Effective communication and presentation skill.
PO4: Ability to understand professional and ethical responsibility.
PO5: Recognition of the need for life-long learning.

Number of practical problems in journal

Unit Number	Number of Questions	Time required to implement and debug the question (in hours)	Minimum required of journal Certification	Date of Certification
1.	1	02	1	4th week
2.	2	06	2	6th week
3.	2	06	2	8th week
4.	5	19	5	10th week
5.	2	10	2	13th week
6.	1	05	1	
Total	**13**	**48**	**13**	**-**

Practical Guidelines:

- Students must prepare handwritten workbook for textual answer and source code .
- In case of image or diagram answer then student enrolment number must be present in photograph/image.
- Students will not be permitted to attend the laboratory unless they bring the practical workbook .
- Students should obtain the signature of the allocated lab teacher in the workbook after completing each practical exercise.
- Script should be well-documented and must follow the Coding standards.
- Workbook will be evaluated based on rubric parameters given below:
  - Completeness of solution
  - Correctness of logic
  - Reusability of code
  - Solution quality
  - Demonstration
  - Technical viva

Tools and technologies:

A student must have the following computing environment in laboratory and or on his/her laptop.

- ✓ PHP 5 or above
- ✓ MYSQL 8.0 or above

Suggested Learning Resources:

1. Steven Holzner, The Complete Reference PHP, Mc Graw Hill.
2. Steve Suehring, Tim Converse and Joyce Park, PHP6 and MySQL, Wiley India Pvt. Ltd.
3. Hugh E. Williams, PHP and MySQL, O'Reilly.
4. Sharanam Shah, Vaishali Shah - Java EE 6 Server Programming, Shroff publishers & Distributors
5. http://php.net/manual/en/
6. http://api.jquery.com/Ajax_Events/
7. https://learn.jquery.com/ajax/jquery-ajax-methods/
8. http://www.w3schools.com/PHP

Develop Contest Management System that allows participant to register themselves in a contest such as Food Making contest, Photo contest, etc. Here Visitors can access basic information regarding available contest. System has three users such as:

1) Contest Coordinator

2) Registered User

3) Visitor

The Contest Coordinator is responsible for maintaining details of Food Making Contest. Contest Coordinator can view the registered participants details as well as manage each participant score given by judges in order to generate contest result.

Visitors can only view the available Food Making Contest details. After registration in a system visitors become registered user and can participate in the available Contest. Also Participant can view their result.

Applicable after Practical No. 5.

Consider the following relations to store the data.

Relations:

Registration (<u>userId</u>, Name, mobileNo, username, password, email, address, city, userRegistrationDate, winnerCount)

Contest (<u>cId</u>, cName, cType, cDate, cDuration, cDescription, contestBanner)

Participation (<u>pId</u>, cId, userId, participantRegistrationDate)

Result (pId, scoreByJudge1, scoreByJudge2, scoreByJudge3, timeTaken, scoreEntryDate)

ContestResult (<u>rId</u>, cId, winnerPid, runnerupPid) Update the value of winnerCount field of Registration table with the help of trigger.

Note: Consider "Contest Management System" with its given database and implement the asked problem. It is mandatory to design a web page using CSS and apply validation on all the fields according to nature of its value. Don't give only required field validation on form fields.

Practical No.: 01	Group: A		Enrolment No.:
Objective:	To be able to implement the learning of data structure available in PHP.		
Problem definition:	Consider the given array that containing the Level of the contest, Contest name, Contest date, and Registration fees in ₹.		

$Contest=array(

     "National"=>array("Gujarati breakfast contest"=>array("04-02-2023",500),"Best breakfast contest"=>array("10-02-2023",700)),

     "International"=>array("Best fast food contest"=>array("07-02-2023",300),"Best cake making contest"=>array("07-02-2023",400)),

     "State"=>array("Gujarati nasta contest"=>array("26-04-2023",1000),"Punjabi dish making contest"=>array("24-04-2023",800)),

     "Local"=>array("South Indian food contest"=>array("12-05-2023",1000),"Crunchy food contest"=>array("05-07-2023",700))

     );

- Add at least another four records in above specified "Contest" array and by considering the array display the details in given format based on user's choice.

  Ex: If user selects "State" as a Level of contest then show available contest details for State level contest in given format on same page.

  26-04-2023

  24-04-2023

  Output:

Select Level of Contest:
○ Local ● State ○ National ○ International

Show

"**State level contest**" details

Name of Contest	Date of Contest	Registration fees (in ₹)
Gujarati nasta contest	26-04-2023	1000
Punjabi dish making contest	24-04-2023	800

Time required to implement and debug: 02 Hours	CO Mapped: CO1		PO Mapped: PO1, PO2		
References to solve the problem:	Steven Holzner, The Complete Reference, Tata McGraw Hill, Page No. 55-74 and 92-104				
Solution must contain:	Source code and sample test data with expected output screenshot				
Evaluation Parameters: *1. Addition of other four values in an array 2. Correct and optimized logic to fetch data dynamically 3. Usage of appropriate condition and looping construct 4. Embedding HTML and PHP 5. Technical Viva*					
Marks:	3	4	3	2	3
Secured by Student:					
Teacher's Signature with Date:					

Practical No.: 02	Group: A		Enrolment No.:	
Objective:	To apply the learning of client-side and server-side validations.			
Problem Definition:	Consider Practical-1 and provide a provision of "Registration" for visitors to become a Register user. Register user can participate in any of the available contests. Take appropriate control to design the registration form and apply client-side and server-side validation on fields based on its nature. Registration form of the participant includes first name, middle name last name, mobile number, email address, username, password, retype password, gender, permanent address, city and two buttons one for signup and other for clear. Also Provide functionality to add temporary address, for that generate address box on the bases of participant choice. 1. Name should not contain any number or special symbol. 2. Password length must be al-least 8 character, and it must have at-least one number and one special symbol. 3. Email should be valid email address. 4. Mobile number contains only number. 5. Password and retype password must be matched. 6. Apply appropriate validations for Address and Gender fields. 7. Whenever user click on clear button data should be clear from all fields. 8. Give provision for subscription so user can receive upcoming contest notification. ☑ Subscribe with us to receive upcoming contest notification 9. According to the definition find and implement at least two fields appropriate for registration form, other than specified in problem definition. Also specify the reason for including those fields. Implement the appropriate validation on same.			
Time required to implement and debug: 03 Hours	**CO Mapped:** CO1 and CO2		**PO Mapped:** PO1, PO2	
References to solve the problem:	Steven Holzner, The Complete Reference PHP, Mc Graw Hill, Page No: 245-260			
Solution must contain:	Source code and sample test data with expected output screenshot.			
Evaluation Parameters: *1. Find, justify and apply validation on two fields 2. Usage of appropriate type of validation 3. Dynamic control generation and constraint management 4. Technical Viva*				
Marks:	**4**	**3**	**5**	**3**

Secured by Student:				
Teacher's Signature with Date:				

Practical No.: 03	Group: A		Enrolment No.:	
Objective:	To acquaintance the concept of state management techniques.			
Problem Definition:	Based on implementation of Practical:02, develop a login page where authenticate the participant and provide remember me facility and display all information of registration form in proper format on another page while clicking on login button. As and when participant wants to logout from the system then provide provision for it. Participants will be logged out from the system automatically after five minutes if he/she is not active. Modify the solution of practical: 02 according to the requirement. – Display Participant username in camel case, in top-middle of the page. i.e. **Welcome <Mr./Ms> < Username>** Title Mr. or Ms. According to participant's gender. – Display last logged in date and time in top right corner of the page. – Email id must be displayed in lower case letter only. – Display address and mobile number in proper format. – Display the number of days remaining to participate in each available contest in proper format. Manage the errors to be generated through custom error handler.			
Time required to implement and debug: 03 Hours	**CO Mapped:** CO1 and CO2		**PO Mapped:** PO1, PO2 and PO4	
References to solve the problem:	Steven Holzner, The Complete Reference PHP, Mc Graw Hill, Page No: 245-260 http://php.net/manual/en/features.cookies.php			
Solution must contain:	Source code and sample test data with expected output.			
Evaluation Parameters: *1. Usage of appropriate state management techniques* 2. *Custom error handler 3.* *Optimized logic for retrieval of data 4. Technical Viva*				
Marks:	**4**	**4**	**4**	**3**
Secured by Student:				
Teacher's Signature with Date:				

Practical No.: 04	Group: B	Enrolment No.:
Objective:	To apply the learnings of file handling techniques in application development.	
Problem Definition:	In connection of Practical-3. When participant check the 'Subscribe with us to receive upcoming contest notification' then provide notification type as given below. Participant can subscribe with both types of notification or any one type of notification of their choice with unique email address and mobile number. **Upcoming Contest Notification:** ☑ Email Notification ☐ Mobile SMS 1. Store participant name, email address, mobile number and subscribed notification type in 'ContestNotification.csv' file. 2. Shall we use any other delimiter other than comma (,) in .csv file? 3. Write the format of 'ContestNotification.csv' file with at least two dummy records with actual values. 4. Check for the file existence otherwise appropriate message will be displayed. 5. Count total subscribed users with both 'Email Notification' and 'Mobile SMS' notification type. 6. Display the most convenient notification type among the participants. 7. Display name, email address and mobile number of participant on web page in formatted form, who have subscribed with 'most convenient notification type'. *Do not use static value for most convenient notification type.*	

Time required to implement and debug: 03 Hours	**CO Mapped:** CO3	**PO Mapped:** PO1, PO2 and PO4
References to solve the problem:	Steven Holzner, The Complete Reference, Tata McGraw Hill, Page No. 55-74 and 92-104	

Evaluation Parameters: *1. Optimized logic for unique email address and mobile number with validation 2. Output in a formatted form 3. Taken care for the file handling parameters 4. Technical Viva*

Marks:	5	2	3	5
Secured by Student:				
Teacher's Signature with Date:				
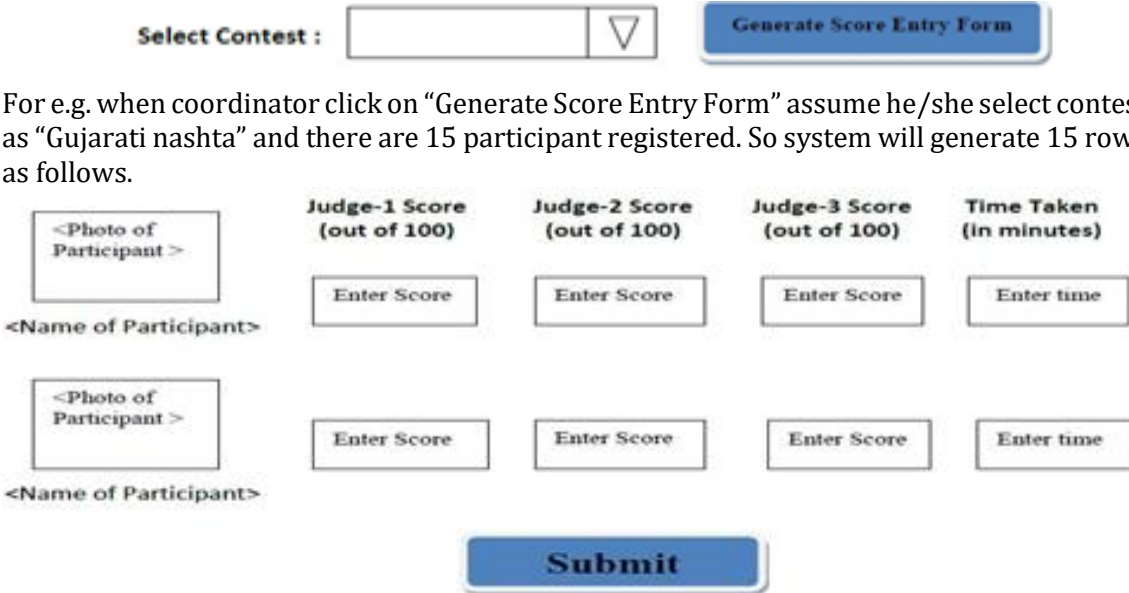
Practical No.: 05	Group: B	Enrolment No.:
Objective:	To implement the file uploading with proper validation in real life application.	
Problem Definition:	Modify the solution of Practical-2 and provide the provision for Participant to upload his/her profile picture and identity proof. As an identity proof participant can upload Aadhar card or Election card.	

1. Profile picture should be maximum 100 KB and only support the file format such as JPEG, JPG and PNG
2. Check identity proof document must be in PDF only and it is of maximum 2 MB.
3. Store the uploaded file on permanent basis at particular location of server.
4. Using script prevent the participant to upload a file with the same name as an existing file in the same folder.
5. Develop Header and Footer file related to Contest Management System and use the same files on relevant pages.
6. Implement the concept of error handling for file upload.

Time required to implement and debug: 02 Hours	**CO Mapped:** CO3	**PO Mapped:** PO1, PO2 and PO4
References to solve the problem:	Steven Holzner, The Complete Reference, Tata McGraw Hill, Page No. 55-74 and 92-104	

Evaluation Parameters: *1. Implementation of validations for filetype and file size 2. Provision to prevent same filename at same location and error handling 3. Usage of repetitive functions in best of manner 4. Technical Viva*

Marks:	5	2	3	5
Secured by Student:				
Teacher's Signature with Date:				

Practical No.: 06	Group: B		Enrolment No.:	
Objective:	To implement dynamic web applications with the help of database connectivity.			
Problem Definition:	1. Consider Practical-2, create and implement the user defined function for user registration functionality. This function will add records in Registration table. **Contest Coordinator** 2. Assume that Contest Coordinator username "coordinator@gmail.com" and password "pass@coordinator" already stored in Registration table. - When Contest Coordinator is authenticated then he/she can view the following menus : **Dashboard | Manage Registered User | Manage Contest | Manage Score Entry | Reports** - In "Manage Registered User", Contest Coordinator can view a list of all registered user in descending order by registration date and Contest Coordinator can activate or deactivate the particular user. Manage Registered User page includes: Name, Mobile number, Email, Address, City, and User registration date. - Implement "Manage Contest", here Contest Coordinator can add new contest by submitting given form. Add contest details in the Contest table on successful submission of the form. **Add Contest** Name: Contest Level: --Select Contest Level-- Date: mm / dd / yyyy Duration (in minutes): About Contest: Upload Contest Banner: Browse... No file ...lected. Submit Clear Use of PreparedStatement is necessary for at least two database operations.			
Time required to implement and debug: 03 Hours	**CO Mapped:** CO2 and CO3		**PO Mapped:** PO1, PO2 and PO4	
References to solve the problem:	Steven Holzner, The Complete Reference, Tata McGraw Hill.			
Evaluation Parameters: *1. User defined function 2. Authentication 3. Display registered user in descending order 4. Database operations using prepared statement 5. Add contest 6. Technical Viva*				

Marks:	2	2	1	3	2	5
Secured by Student:						
Teacher's Signature with Date:						

Practical No.: 07	Group: B	Enrolment No.:
Objective:	To implement the learning of structured data retrieval and processing in dynamic web application development.	
Problem Definition:	**Registered User** 1. In the continuity of Practical-6, modify the script for login functionality of Practical-3 and implement the same with the help of database connectivity. After successful authentication Registered user shall have following menus: **Home	Update Profile	Contest	Participation History** 2. By clicking on "Contest" menu Registered user can view each available contest in following given format. Also by clicking on Title of contest or Banner of contest user shall view detail about that contest. 	

<div align="center">

Contest-1   **Contest-2**

< Title of Contest > <Level of contest>   **< Title of Contest > <Level of contest>**

<Date of contest>   *<Date of contest>*

<Banner of Contest >   **<Banner of Contest >**

Click here to Register   **Click here to Register**

</div>

3. Registered users can participate in any of the available contests before at least two days of the contest organized date of the respective contest. Registered users need to click on "Click here to Register" button to participate in the contest.
4. After successful participation of the user, displays the participation information in a proper format. Participation information includes participant name, participation date, contest name, contest date, and no. of days remain to start the contest.
5. At upper right corner of a web page, the time of login into the system will be displayed.

	6. Develop a web page to update registered user's profile where he/she can change their personal details as well as the theme according to their choice. Update registered user's profile page display auto-filled registration from where customer can make changes.
Time required to implement and debug: 04 Hours	**CO Mapped:** CO2 and CO3	**PO Mapped:** PO1, PO2 and PO4

References to solve the problem:	Steven Holzner, The Complete Reference, Tata McGraw Hill.

Evaluation Parameters: *1. Display required details using stored procedure 2. Optimized logic to verify contest registration date. 3. Contest registration functionality. 4. Presentation of participation information 5. Update user's profile 6. Technical Viva*

Marks:	2	2	3	2	1	5
Secured by Student:						
Teacher's Signature with Date:						

Practical No.: 08	Group: B	Enrolment No.:
Objective:	To implement the learning of data analysis in web application development.	
Problem Definition:	1. Make appropriate changes in practical-7 to allow Register user to participate only in two contests in a month. If a user tries to register in the third contest in the same month then show an appropriate message to the user. 2. Registered user can view his/her own participation history of past one year by clicking on "Participation History" menu. 3. Create and implement a function to allow a registered user to participate in a third contest in month. At the time of registering in the third contest, four-digit random numbers will be generated as a participant code if the registered user had participated in at least one contest organized in the last three months. The user needs to provide that code to register in the third contest.	
Time required to implement and debug: 03 Hours	**CO Mapped:** CO2 and CO3	**PO Mapped:** PO1, PO2 and PO4
References to solve the problem:	Steven Holzner, The Complete Reference, Tata McGraw Hill.	
Evaluation Parameters: *1. Business logic to allow Registered user to participate only in two contests in a month 2. Four-digit random number generation 3. Technical Viva*		
Marks:	5	5	5
Secured by Student:			
Teacher's Signature with Date:			

Practical No.: 09	Group: B	Enrolment No.:
Objective:	To implement the data representation functionality in dynamic web application using structured data.	
Problem Definition:	**Contest Coordinator**	

Contest Coordinator

1. Develop a web page for Contest Coordinator dashboard with given functionality, where Contest Coordinator can view contest wise participant list. Also display count of total participants and total online users. The page will atomically get refreshed at every 20 seconds.

2. By clicking on "Manage Score Entry" menu Contest Coordinator can do the entry of each participant score given by judges in order to generate particular contest result as per following.

Contest score entry:

For e.g. when coordinator click on "Generate Score Entry Form" assume he/she select contest as "Gujarati nashta" and there are 15 participant registered. So system will generate 15 rows as follows.

When coordinator click on "Submit" button result will be stored in database and the result page is displayed as follow. Increase winner count of winner participant in Registration table with the help of trigger only.

Contest result:

Contest Result

<<Contest Name>>
<<Contest Date>> <Level of Contest>>

Winner:

Name	Total Score	Average Score	Time Taken
<<Name of Winner >> <Photo of Winner Participant >	<<Total Score>>	<<Average Score>>	<<Time taken>>

Runner up:

Name	Total Score	Average Score	Time Taken
<<Name of Runner up >> <Photo of Runner up Participant >	<<Total Score>>	<<Average Score>>	<<Time taken>>

<<Date and time of result generation >>

Time required to implement and debug: 04 Hours	CO Mapped: CO2 and CO3		PO Mapped: PO1, PO2 and PO4		
References to solve the problem:	Steven Holzner, The Complete Reference, Tata McGraw Hill.				
Evaluation Parameters: *1. Dashboard with required data 2. Score entry 3. Implementation of trigger 4. Contest result 5. Technical Viva*					
Marks:	4	2	3	3	3
Secured by Student:					
Teacher's Signature with Date:					

Practical No.: 10	Group: B		Enrolment No.:		
Objective:	To generate summary reports from structured data in a dynamic web application.				
Problem Definition:	Contest Coordinator can generate various kind of reports as follows: I. Contest Coordinator can view Top two participant details who have secure maximum winner count . II. Contest Coordinator can view list of top ten areas (city), from where maximum participant have participated in various contest. III. List top five contest which have maximum participation. IV. Provide facility to Contest Coordinator that he/she can able to generate to date to from date contest organized report. V. Contest Coordinator can view contest wise winner and runner up detail report.				
Time required to implement and debug: 03 Hours	**CO Mapped:** CO2 and CO3		**PO Mapped:** PO1, PO2 and PO4		
References to solve the problem:	Steven Holzner, The Complete Reference, Tata McGraw Hill.				
Evaluation Parameters: *1. Display top two participant details who have secure maximum winner count 2. List top ten areas for maximum* participant *3. Top five contests which have maximum participation 4. Date wise report 5. Technical Viva*					
Marks:	**4**	**3**	**3**	**3**	**2**
Secured by Student:					
Teacher's Signature with Date:					

Practical No.: 11	Group: B				Enrolment No.:	
Objective:	To provide the knowledge of asynchronous data transmission.					

Problem Definition:	1. Consider Practical-9 and modify the code according to requirement.
	  • Load participants score entry for selected contest without loading the whole web page.
	  • Display appropriate loader for the web page when it makes a request to server till to get a response.
	
	2. Implement CAPTCHA facility in practical: 02 using AJAX and JQuery, to ensure that the form is submitted by human intervention only and there is no involvement of any tools or program.
	
	3. Consider Practical-7 and give any hovered effect on "Contest banner" and "Click here to Register" button return it to normal if the participant is no longer hovering.
	
	4. Display the two "Contest title" which had the highest number of participants in the year 2020, By clicking on the title 'registered users' user can view the detail about selected contest including winner and runner-up participant's details. Give toggle (Show Detail and hide detail on the same page) effect on Contest details whenever a specific contest title is clicked.					

Time required to implement and debug: 05 Hours	**CO Mapped:** CO4			**PO Mapped:** PO1, PO2 and PO4		
References to solve the problem:	https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/jQuery_Basics.html http://api.jquery.com/Ajax_Events/					
Evaluation Parameters: *1. Data loaded in Asynchronous manner 2. Usage of loader 3. Captcha using JQuery 4. Hovered effect on Image 5. Toggle effect on service name 6. Technical Viva*						
Marks:	**4**	**2**	**3**	**1**	**2**	**3**
Secured by Student:						
Teacher's Signature with signature:						

Practical No.: 12	Group: B		Enrolment No.:	
Objective:	To provide hands-on practice regarding data parsing using XML and JSON.			

Problem Definition:	1. Consider the ContestNotification.csv' file created in Practical-04 and convert the content of the file into JSON format such as given sample record,
	
	Sample record in JSON Format
	`{`
	`"Participant Name":"Pritesh Patel",`
	`"EmailAddress": "pritesh10@gmail.com",`
	`"NotificationType ":[" Email Notification","Mobile SMS"]`
	`}`
	
	2. Write one significant difference between JSON Object and JSON Array. Write a code to access JSON data using both methods.
	3. Sort above JSON data by participant name and display on console.
	4. Consider the practical 7 to 9 and display the contest wise number of participants during year 2020 in graphical format.			

Time required to implement and debug: 05 Hours	**CO Mapped:** CO4		**PO Mapped:** PO1, PO2 and PO4	
References to solve the problem:	https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/jQuery_Basics.html http://api.jquery.com/Ajax_Events/			
Evaluation Parameters: *1. Conversion into JSON format 2. Valid description for JSON Object and JSON Array with accessing example 3. Sorted data display on console 4. Display in graphical form 5. Technical Viva*				
Marks:	3	3	2	4	3
Secured by Student:					
Teacher's Signature with Date:					

Practical No.: 13	Group: B		Enrolment No.:
Objective:	To provide insights regarding creation of custom web services as well as usage of readily available APIs.		
Problem Definition:	Consider Practical-03, 09 and create a web service for Sign Up, Sign In process and displaying Contest result details. Consider the following directory structure and develop web service for signup, sign in for Registered user and displaying Contest result details. API |-----Config/ |------------database.php, for the purpose of database connectivity. |-----Objects/ |------------RegisteredUser.php, it contains the appropriate methods like signUp(), signIn() |------------Result.php it must it contains the appropriate methods like contestResult() |-----User/ |------------Signup.php, it will accept user data and save into the Registration table. |------------Signin.php, it will accept username, password and authenticate the Registered user . |------------Contestresult.php, it will accept contest id and return winner and runner up participant details.		
Time required to implement and debug: 05 Hours	CO Mapped: CO4, CO5		PO Mapped: PO1, PO2 and PO4
References to solve the problem:	https://www.codeofaninja.com/2017/02/create-simple-rest-api-in-php.html		
Evaluation Parameters: *1. Implemented given directory structure 2. Usage of session management 3. Sign Up and Sign In 4. Displaying Contest result 5. Technical Viva*			

Marks:	2	2	4	4	3
Secured by Student:					
Teacher's Signature with Date:					