**VASAVI COLLEGE OF ENGINEERING (AUTONOMOUS)**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**WEB TECHNOLOGIES**

**ASSIGNMENT – I**

**Instructions**

1. Use Brackets software for your assignment, install html validator and beautify plugins and use them for debugging and indentation.

2. Make sure that there are no warnings or errors in your assignment.

3. All css and javascript code must be written in external files (i.e .css and .js files seperately)

4. Make sure that all the source code (.html, .css and .js files ) has appropriate comments containing your roll number and name.

5. Use the appropriate front end framework given in the question.

6. You must validate all the input fields present in the ‘Contact Us/Login/Registration” web page using JavaScript.

The ‘Contact Us/Login/Registration’ web page must have at least 3 fields i.e name, email id and mobile number.

7. Submit the assignment on or before 15th February 2019 as a single .zip file containing the (.html, .css and .js files as well as images). Below are instructions to submit your assignment

i) Email the assignment to: [assignment.it.b@gmail.com](mailto:assignment.it.b@gmail.com)

ii) Use Subject as: Web Technologies: Assignment 1: Your Full roll number

E.g Web Technologies: Assignment 1: 1602-16-737-061

iii) The name of the .zip file must be your\_roll\_number.zip

E.g 1602\_16\_737\_061.zip

8. Do not copy from others; copying will result in allocation of 0 marks.

9. The assignment questions are present in WT\_Questions.zip

10. Next page contains assignment question allocated to each student.

**VASAVI COLLEGE OF ENGINEERING (AUTONOMOUS)**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**WEB TECHNOLOGIES**

**ASSIGNMENT – II**

**Extension of Assignment-I**

**Instructions**

1. Use Brackets software or Notepad++ for your assignment; install python or Java latest versions, Xampp to implement your application.

2. Make sure that there are no warnings or errors in your assignment.

3. All css and javascript, Serverside(either python,sevrlets,JSP)code must be written in external files (i.e .css, .js, .java, .py files seperately)

4. Make sure that all the source code (.html, .css, .js, .java, .py files ) has appropriate comments containing your roll number and name.

5. Use the appropriate server side programming (Python,Java Servlets,JSP) given in the question.

6. You must validate all the input fields present in the Login/Registration” web page using JavaScript. If there is no Login or Registration section try to incorporate these pages in your application.

7. Submit the assignment on or before 5th April 2019 as a single .zip file containing the (.html, .css, .js , .java , .py files as well as images). Below are instructions to submit your assignment

i) Email the assignment to: [assignment.it.b@gmail.com](mailto:assignment.it.b@gmail.com)

ii) Use Subject as: Web Technologies: Assignment II: Your Full roll number

E.g Web Technologies: Assignment II: 1602-16-737-061

iii) The name of the .zip file must be your\_roll\_number.zip

E.g 1602\_16\_737\_061.zip

8. Do not copy from others; copying will result in allocation of 0 marks.

9. The assignment questions are present in WT\_Questions.zip

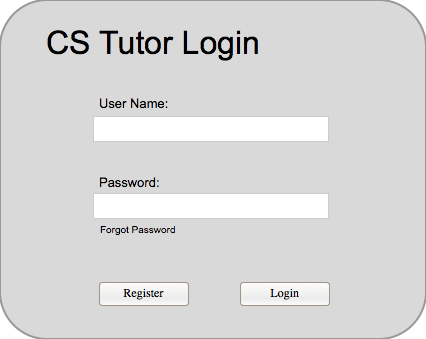
10. Next page contains assignment question allocated to each student.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Roll No** | **Question No** | **Roll No** | **Question No** | **Roll No** | **Question No** |
| 1602-16-737-062 | 01( Python) | 1602-16-737-101 | 24(JSP) | 1602-16-737-094 | 17(Servlets) |
| 1602-16-737-065 | 02(Servlets) | 1602-16-737-103 | 25( Python) | 1602-16-737-095 | 18(JSP) |
| 1602-16-737-066 | 03(JSP) | 1602-16-737-105 | 26(Servlets) | 1602-16-737-104 | 19( Python) |
| 1602-16-737-067 | 04( Python) | 1602-16-737-111 | 27(JSP) | 1602-16-737-106 | 20(Servlets) |
| 1602-16-737-068 | 05(Servlets) | 1602-16-737-113 | 28( Python) | 1602-16-737-107 | 21(JSP) |
| 1602-16-737-069 | 06(JSP) | 1602-16-737-118 | 29(Servlets) | 1602-16-737-109 | 22( Python) |
| 1602-16-737-073 | 07( Python) | 1602-16-737-119 | 30(JSP) | 1602-16-737-110 | 23(Servlets) |
| 1602-16-737-074 | 08(Servlets) | 1602-16-737-120 | 01( Python) | 1602-16-737-112 | 24(JSP) |
| 1602-16-737-075 | 09(JSP) | 1602-16-737-314 | 02(Servlets) | 1602-16-737-114 | 25( Python) |
| 1602-16-737-079 | 10( Python) | 1602-16-737-316 | 03(JSP) | 1602-16-737-116 | 26(Servlets) |
| 1602-16-737-081 | 11(Servlets) | 1602-16-737-318 | 04( Python) | 1602-16-737-117 | 27(JSP) |
| 1602-16-737-082 | 12(JSP) | 1602-16-737-320 | 05(Servlets) | 1602-16-737-317 | 28( Python) |
| 1602-16-737-083 | 13( Python) | 1602-16-737-321 | 06(JSP) | 1602-16-737-319 | 29(Servlets) |
| 1602-16-737-085 | 14(Servlets) | 1602-16-737-322 | 07( Python) |  |  |
| 1602-16-737-087 | 15(JSP) | 1602-16-737-063 | 08(Servlets) |  |  |
| 1602-16-737-091 | 16( Python) | 1602-16-737-076 | 09(JSP) |  |  |
| 1602-16-737-092 | 17(Servlets) | 1602-16-737-077 | 10( Python) |  |  |
| 1602-16-737-093 | 18(JSP) | 1602-16-737-080 | 11(Servlets) |  |  |
| 1602-16-737-096 | 19( Python) | 1602-16-737-084 | 12(JSP) |  |  |
| 1602-16-737-097 | 20(Servlets) | 1602-16-737-086 | 13( Python) |  |  |
| 1602-16-737-098 | 21(JSP) | 1602-16-737-088 | 14(Servlets) |  |  |
| 1602-16-737-099 | 22( Python) | 1602-16-737-089 | 15(JSP) |  |  |
| 1602-16-737-100 | 23(Servlets) | 1602-16-737-090 | 16( Python) |  |  |

### User Login-Registration Functional Requirements

The following image depicts what the login screen looks like to any user. The login screen allows registered users to login to the site to access all of the features that their account gives them access to. If they type in their username and password and click submit the user’s credentials are validated (checked in the database) and if correct they are logged in.

**The empty login dialog.**

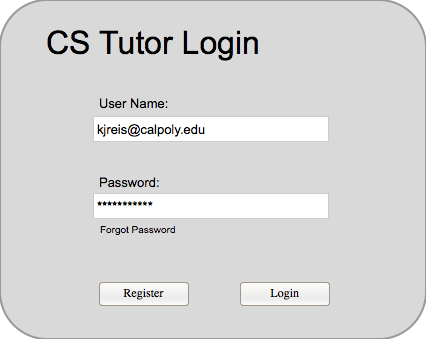


**Validations**: 1) Username could be unique to every user; you can give email id as username also

2) Password length could be minimum eight characters and it should be a combination of at least one digit, one special character, one uppercase, one lowercase

3) Username and password should not be empty

**The filled login dialog.**



Validations:1)Check in the entered user details are authentic or not by interacting with Database

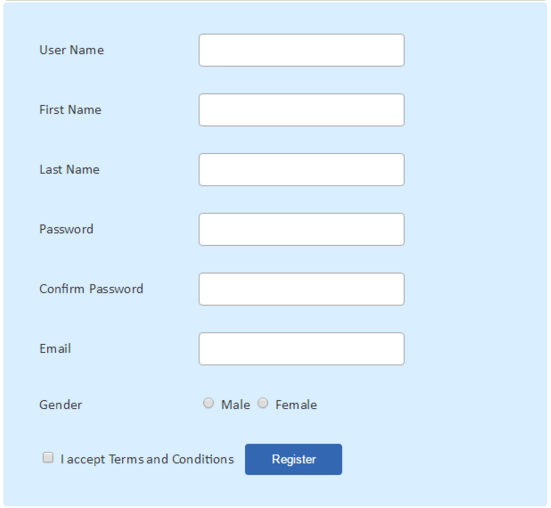
If they are incorrect they get an error message as seen below.

**The incorrect credentials dialog.**



If the user has forgotten their password they click "Forgot Password?" which takes them to a password recovery screen. If the user does not have an account then they click the register button and be taken to the registration screen seen below.

**The empty user registration dialog**



In the registration screen a new user types in all of their information and clicks submit, the data is then validated to make sure there is not an existing user with those credentials. If there is an existing user then the user is asked to enter a new username. If there is no conflict with the credentials then the user is registered and are sent an email to verify registration.

If the user clicked "Forgot Password?" on the login screen then they see the below dialog. After the user types in their email address and clicks submit the email address is checked to see if it belongs to a registered user and if it does then an email with a secure password reset token is emailed to the user.

**The empty password rest dialog.** **The filled password reset dialog.**