**Computer Networks**

**Practical File**

**Name: SHIVA YADAV**

**College Roll No: 206441**

**Examination Roll No: 20035582043**

**Course: BSc Physical Science**

**Sem: VI**

**HTML Practicals**

Ques1: Write a HTML program to design a form which should allow to enter your

personal data.

Solution:  
Code:

<html>

    <body>

    <form>

        <h3 style="text-align:center;"> Enter the following details </h3>

        <table>

            <tr>

                <td>First Name:</td>

                <td><input type="text" ></td>

            </tr>

            <tr>

                <td>Last Name:</td>

                <td><input type="text"></td>

            </tr>

            <tr>

                <td>Mobile Number:</td>

                <td><input type="tel"></td>

            </tr>

            <tr>

                <td>Email:</td>

                <td><input type="email"></td>

            </tr>

            <tr>

                <td>Password:</td>

                <td><input type="password"></td>

            </tr>

            <tr>

                <td>Gender:</td>

                <td>

                    <input type="radio" name="gender">Male

                    <input type="radio" name="gender">Female

                    <input type="radio" name="gender">Others

                </td>

            </tr>

            <tr>

                <td>Marital Status:</td>

                <td>

                    <select>

                        <option>Married</option>

                        <option>Un-Married</option>

                    </select>

                </td>

            </tr>

            <tr>

                <td>Education Qualification:</td>

                <td>

                    <input type="checkbox">High School

                    <input type="checkbox">Undergraduate

                    <input type="checkbox">Postgraduate

                    <input type="checkbox">PhD

                </td>

            </tr>

        </table>

        Personal Details Taken

        <ol>

                <li>First and Last Name</li>

                <li>Mobile No.</li>

                <li>Email ID and Password</li>

                <li>Gender</li>

                <li>Marital Status</li>

                <li>Euducation Qualification</li>

        </ol>

        <input type="submit" value="Submit">

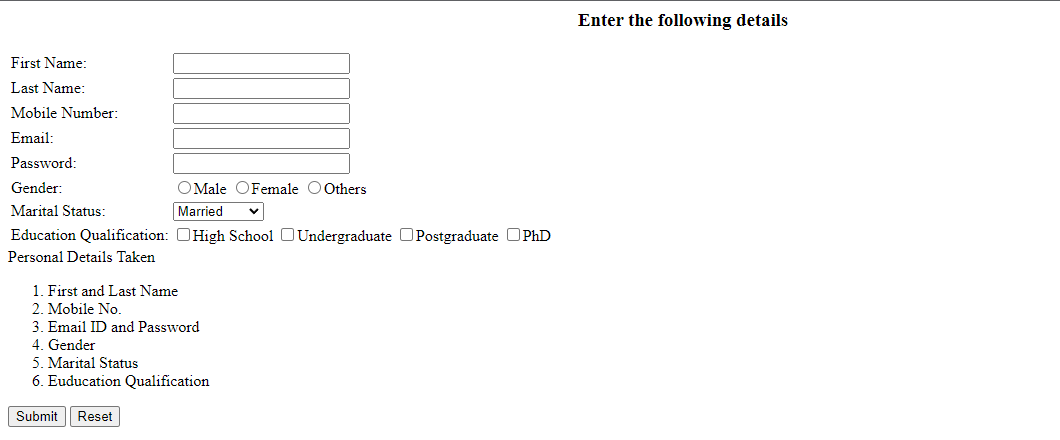
        <input type="reset" value="Reset">

    </form>

    </body>

</html>

Output:



Ques2: Write html code to generate following output.

• Coffee  
• Tea  
 o Black Tea  
 o Green Tea   
• Milk

Solution:  
Code:

<html>

    <ul>

        <li>Coffee</li>

        <li>Tea</li>

        <ul>

            <li>Black Tea</li>

            <li>Green Tea</li>

        </ul>

        <li>Milk</li>

</html>

Output:  


Ques3: Design an html form to take the information of a customer visiting a departmental store such as name, contact phone no, preferred days of purchasing, favourite item (to be selected from a list of items), suggestions etc. One should provide button to Submit as well as Reset the form contents.

Solution:  
Code:

<html>

    <body>

        <h1 style="text-align:center;">

            Deparmental Store Login Details

        </h1>

        <form>

            <table>

                <tr>

                    <td>Name:</td>

                    <td><input type="text"></td>

                </tr>

                <tr>

                    <td>Mobile No.:</td>

                    <td><input type="tel"></td>

                </tr>

                <tr>

                    <td>Preferred Days:</td>

                </tr>

                <tr>

                    <td>From:<input type="date"></td>

                    <td>To:<input type="date"></td>

                </tr>

                <tr>

                    <td>Favorite Item:</td>

                    <td>

                        <select>

                            <option>Coca Cola</option>

                            <option>Pepsi</option>

                            <option>Thums Up</option>

                            <option>Limca</option>

                            <option>Fanta</option>

                            <option>Sprite</option>

                        </select>

                    </td>

                </tr>

            </table>

            <td>Suggestions:</td><br>

            <td><textarea rows=4 cols=50></textarea></td><br>

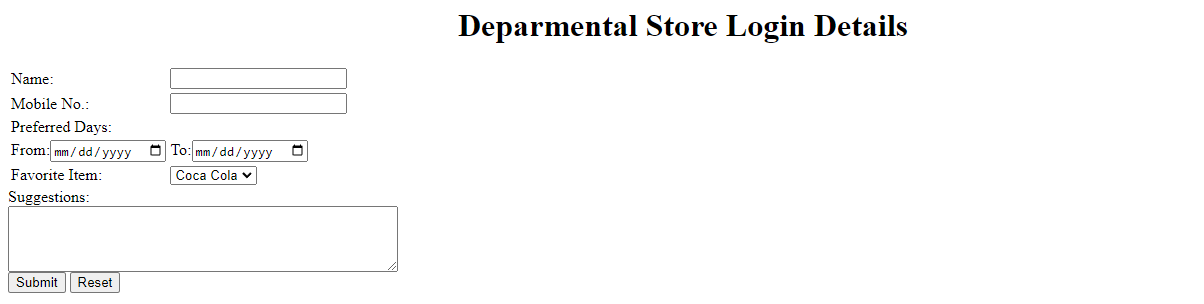
            <input type="submit" value="Submit">

            <input type="reset" value="Reset">

        </form>

    </body>

</html>

Output:  


Ques 4: Design an html form to take the information of an article to be uploaded such as file path, author name, type (technical, literary, general), subject topic (to be selected from a list) etc. One should provide button to Submit as well as Reset the form contents.

Solution:

Code:

<html>

    <body>

        <h1 style="text-align:center;">Article Details</h1>

        <form>

        <table>

            <tr>

                <td>Article Name:</td>

                <td><input type="text"></td>

            </tr>

            <tr>

                <td>Author Name:</td>

                <td><input type="text"></td>

            </tr>

            <tr>

                <td>Article Type:</td>

                <td>

                    <input type="radio" id="tech" name="atype">

                    <label for=tech>Technical</label>

                    <input type="radio" id="lit" name="atype">

                    <label for="lit">Literary</label>

                    <input type="radio" id="gen" name="atype">

                    <label for="gen">General</label>

                </td>

            </tr>

            <tr>

                <td>Subject</td>

                <td>

                    <select>

                        <option>Art and Craft</option>

                        <option>Business</option>

                        <option>Academic Content</option>

                        <option>Entertainment</option>

                        <option>News</option>

                    </select>

                </td>

            </tr>

            <tr>

                <td>Article</td>

                <td><input type="file"></td>

            </tr>

        </table>

        <input type="submit" value="Submit">

        <input type="reset" value="Reset">

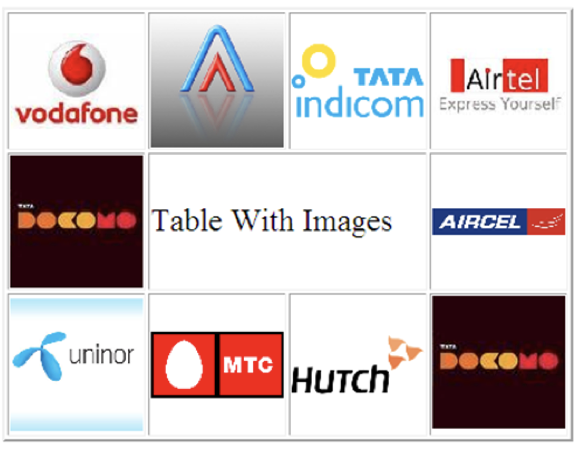
        </form>

    </body>

</html>

Output:  


Ques5: Design an HTML document using Table related tags align the images.



Solution:  
Code:

<html>

    <body>

        <table border="1">

            <tr>

                <td><img src="img1.jpeg" width=90 height=100></td>

                <td><img src="img2.jpeg" width=90 height=100></td>

                <td><img src="img3.jpeg" width=90 height=100></td>

                <td><img src="img4.jpeg" width=90 height=100></td>

            </tr>

            <tr>

                <td><img src="img5.jpeg" width=90 height=100></td>

                <td colspan=2 style="font-size:140%;">Table with Images</td>

                <td><img src="img6.jpeg" width=90 height=100></td>

            </tr>

            <tr>

                <td><img src="img7.jpeg" width=90 height=100></td>

                <td><img src="img8.jpeg" width=90 height=100></td>

                <td><img src="img9.jpeg" width=90 height=100></td>

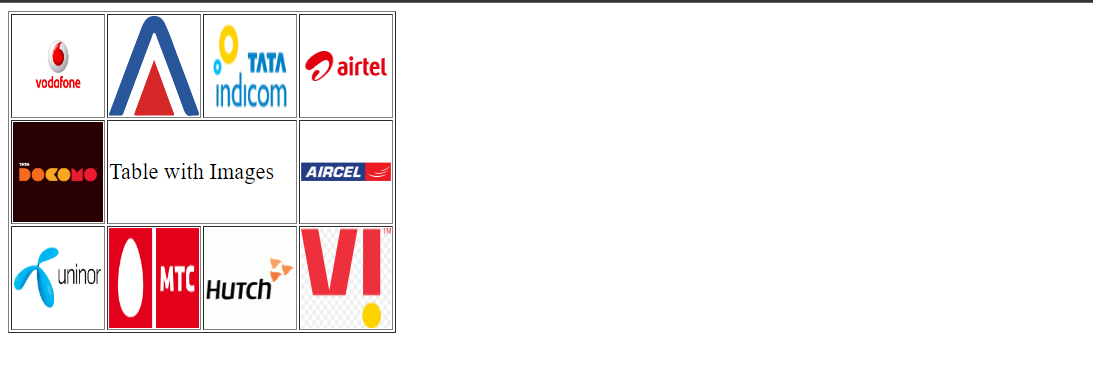
                <td><img src="img10.jpeg" width=90 height=100></td>

            </tr>

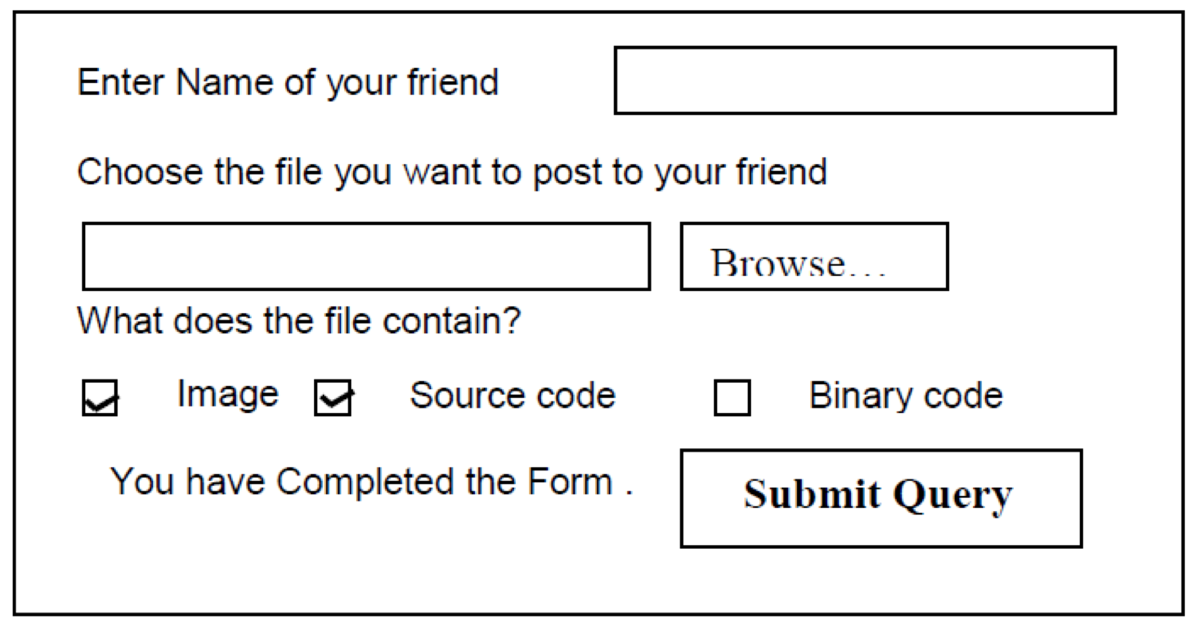
        </table>

    </body>

</html>

Output:  


Ques6: Write a HTML code to generate following output.



Solution:  
Code:

<html>

    <body>

        <form>

            Enter Name of your friend&ensp;&emsp;

            <input type="text">

            <br>

            Choose the file you want to post to your friend

            <br>

            <input type="text"> &emsp;&emsp;<input type="file">

            <br>

            What does the file contain?

            <br>

            <input type="checkbox" id="img">

            &emsp;

            <label for="img">Image</label>

            &emsp;

            <input type="checkbox" id="src">

            &emsp;

            <label for="src">Source Code</label>

            &emsp;

            <input type="checkbox" id="bin">

            &emsp;

            <label for="bin">Binary Code</label>

            <br>

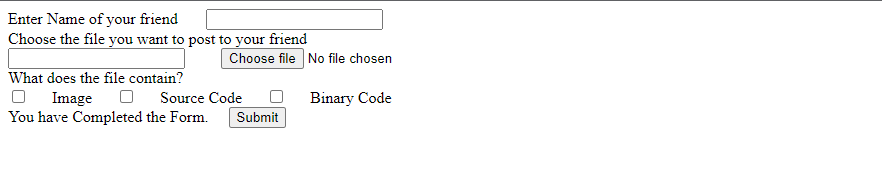
            You have Completed the Form.&emsp;

            <input type="submit">

        </form>

    </body>

</html>

Output:  


Ques7: Develop static pages (using only HTML) of an online Book store.The website should consist of following pages.

Home page  
Registration and user Login  
User profile page  
Books catalog  
Shopping cart  
Payment by credit card Order Conformation

Solution:

Code:

Home Page:

<html>

    <body style="background-color:lightgray;">

        <form style="text-align:center" action="ques7LoginPage.html">

            <h1>Online Book Store</h1>

            <h3>Shop for <b>Books</b>, <b>Comics</b>, <b>Magazine</b> and much more</h3>

            <input type="submit" value="Enter the Store">

        </form>

    </body>

</html>

Login Page:

<html>

    <body style="background-color:lightgray;">

        <button type="button"><a href="ques7HomePage.html">Home</a></button>

        <h1 style="text-align:center;">Online Book Store</h1>

        <form action="ques7InventoryPage.html">

            <table>

                <tr>

                    <td>Username: </td>

                    <td><input type="text" placeholder="Username" required></td>

                </tr>

                <tr>

                    <td>Password: </td>

                    <td><input type="password" placeholder="Password" required></td>

                </tr>

            </table>

            <input type="submit" value="Login">

            <br>

            <a href="ques7RegisterPage.html">New user? Click here.</a>

        </form>

    </body>

</html>

Register Page:

<html>

    <body style="background-color:lightgray;">

        <button type="button"><a href="ques7HomePage.html">Home</a></button>

        <h1 style="text-align:center;">Online Book Store</h1>

        <form action="ques7LoginPage.html">

            <table>

                <tr>

                    <td>Name: </td>

                    <td><input type="text" required></td>

                </tr>

                <tr>

                    <td>Mobile No.: </td>

                    <td><input type="text" pattern="^[0-9]\*$" minlength=10 maxlength=10 required></td>

                </tr>

                <tr>

                    <td>Email: </td>

                    <td><input type="email" required></td>

                </tr>

                <tr>

                    <td>Gender: </td>

                    <td>

                        <input type="radio" id="male" name="gender" required>

                        <label for="male">Male</label>

                        &emsp;

                        <input type="radio" id="female" name="gender" required>

                        <label for="female">Female</label>

                    </td>

                </tr>

                <tr>

                    <td>Interests: </td>

                    <td>

                        <input type="checkbox" id="phy" name="interest">

                        <label for="phy">Physics</label>

                        &emsp;

                        <input type="checkbox" id="cs" name="interest">

                        <label for="cs">Computer Science</label>

                        &emsp;

                        <input type="checkbox" id="math" name="interest">

                        <label for="math">Mathematics</label>

                    </td>

                </tr>

                <tr>

                    <td>

                        <label for="tnc">Accept Terms and Conditions</label>

                        <input type="checkbox" id="tnc" required>

                    </td>

                </tr>

            </table>

            <input type="submit" value="Register">

        </form>

    </body>

</html>

Catalog Page:

<html>

    <body style="background-color:lightgray;">

        <button type="button"><a href="ques7HomePage.html">Home</a></button>

        <button type="button"><a href="ques7UserProfile.html">Profile</a></button>

        <h1 style="text-align:center;">Online Book Store</h1>

        <h2>Book Catalog</h2>

        <ul>

            <li>

                <b>Computer Science</b>

                <ol>

                    <li>The C Programming Language - Brian W. Kernighan, Dennis M. Ritchie</li>

                    <li>Computer Networks - Tanenbaum Andrew S., Wetherall David J.</li>

                    <li>Algorithm Design - Jon Kleinberg, Eva Tardos</li>

                    <li>Compilers Principles Techniques And Tools - Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman</li>

                </ol>

            </li>

            <li>

                <b>Mathematics</b>

                <ol>

                    <li>Elements of discrete mathematics - Liu C. L.</li>

                    <li>Graph theory - Adrian Bondy, U.S.R. Murty</li>

                </ol>

            </li>

            <li>

                <b>Physics</b>

                <ol>

                    <li>Elementary Solid State Physics Principles and Applications - A. Omar.</li>

                </ol>

            </li>

        </ul>

        <br>

        <form action="ques7ShoppingCart.html">

            <input type="submit" value="Shopping Cart">

        </form>

    </body>

</html>

User Profile Page:

<html>

    <body style="background-color:lightgray;">

        <button type="button"><a href="ques7HomePage.html">Home</a></button>

        <button type="button"><a href="ques7InventoryPage.html">Catalog</a></button>

        <h1 style="text-align:center;">Online Book Store</h1>

        <h2>Profile</h2>

        <form action="ques7InventoryPage.html">

            <table>

                <tr>

                    <td>Name: </td>

                    <td><input type="text" placeholder="David"></td>

                </tr>

                <tr>

                    <td>Mobile No.:</td>

                    <td><input type="text" placeholder="999-9999-999" pattern="^[0-9]\*$" minlength=10 maxlength=10></td>

                </tr>

                <tr>

                    <td>Email: </td>

                    <td><input type="email" placeholder="david@example.com"></td>

                </tr>

                <tr>

                    <td><input type="submit" value="Save Changes"></td>

                </tr>

            </table>

        </form>

    </body>

</html>

Shopping Cart Page:

<html>

    <body style="background-color:lightgray;">

        <button type="button"><a href="ques7HomePage.html">Home</a></button>

        <button type="button"><a href="ques7InventoryPage.html">Catalog</a></button>

        <h1 style="text-align:center;">Online Book Store</h1>

        <h2>Shopping Cart</h2>

        <form action="ques7Payment.html">

            <table>

                <tr>

                    <td><b>Book Name</b></td>

                    <td><b>Qty</b></td>

                </tr>

                <tr>

                    <td>The C Programming Language</td>

                    <td>&emsp;<input type="number" value="0" min="0" max="4"></td>

                </tr>

                <tr>

                    <td>Computer Networks</td>

                    <td>&emsp;<input type="number" value="0" min="0" max="4"></td>

                </tr>

                <tr>

                    <td>Algorithm Design</td>

                    <td>&emsp;<input type="number" value="0" min="0" max="4"></td>

                </tr>

                <tr>

                    <td>Compilers Principles Techniques And Tools</td>

                    <td>&emsp;<input type="number" value="0" min="0" max="4"></td>

                </tr>

                <tr>

                    <td>Elements of discrete mathematics</td>

                    <td>&emsp;<input type="number" value="0" min="0" max="4"></td>

                </tr>

                <tr>

                    <td>Graph theory</td>

                    <td>&emsp;<input type="number" value="0" min="0" max="4"></td>

                </tr>

                <tr>

                    <td>Elementary Solid State Physics Principles and Applications</td>

                    <td>&emsp;<input type="number" value="0" min="0" max="4"></td>

                </tr>

            </table>

            <input type="submit" value="Proceed to payment">

        </form>

    </body>

</html>

Payment Page:

<html>

    <body style="background-color:lightgray;">

        <button type="button"><a href="ques7HomePage.html">Home</a></button>

        <button type="button"><a href="ques7InventoryPage.html">Catalog</a></button>

        <h1 style="text-align:center;">Online Book Store</h1>

        <h2 style="text-align:center;">Payment Options</h2>

        <form action="ques7Conformation.html">

            <table>

                <tr>

                    <td>

                        <select>

                            <option>Credit Card</option>

                            <option>Debit Card</option>

                        </select>

                    </td>

                </tr>

                <tr>

                    <td>Card Number: </td>

                    <td><input type="text" pattern="^[0-9]\*$"  placeholder="xxxx-xxxx-xxxx-xxxx" minlength=16 maxlength=16 required></td>

                </tr>

                <tr>

                    <td>Expiry Date: </td>

                    <td><input type="date" required></td>

                </tr>

                <tr>

                    <td>CVV: </td>

                    <td><input type="text" pattern="^[0-9]\*$" placeholder="xxx" minlength=3 maxlength=3 required></td>

                </tr>

                <tr>

                    <td>Card Holder Name: </td>

                    <td><input type="text" placeholder="Name" required></td>

                </tr>

            </table>

            <input type="submit" value="Proceed">

        </form>

    </body>

</html>

Conformation Page:

<html>

    <body style="background-color:lightgray;text-align:center;">

        <h1>Online Book Store</h1>

        <h2>Transaction Completed</h2>

        <br>

        Thank you for using <b>Online Book Store</b>

        <br>

        <button type="button"><a href="ques7HomePage.html">Home</a></button>

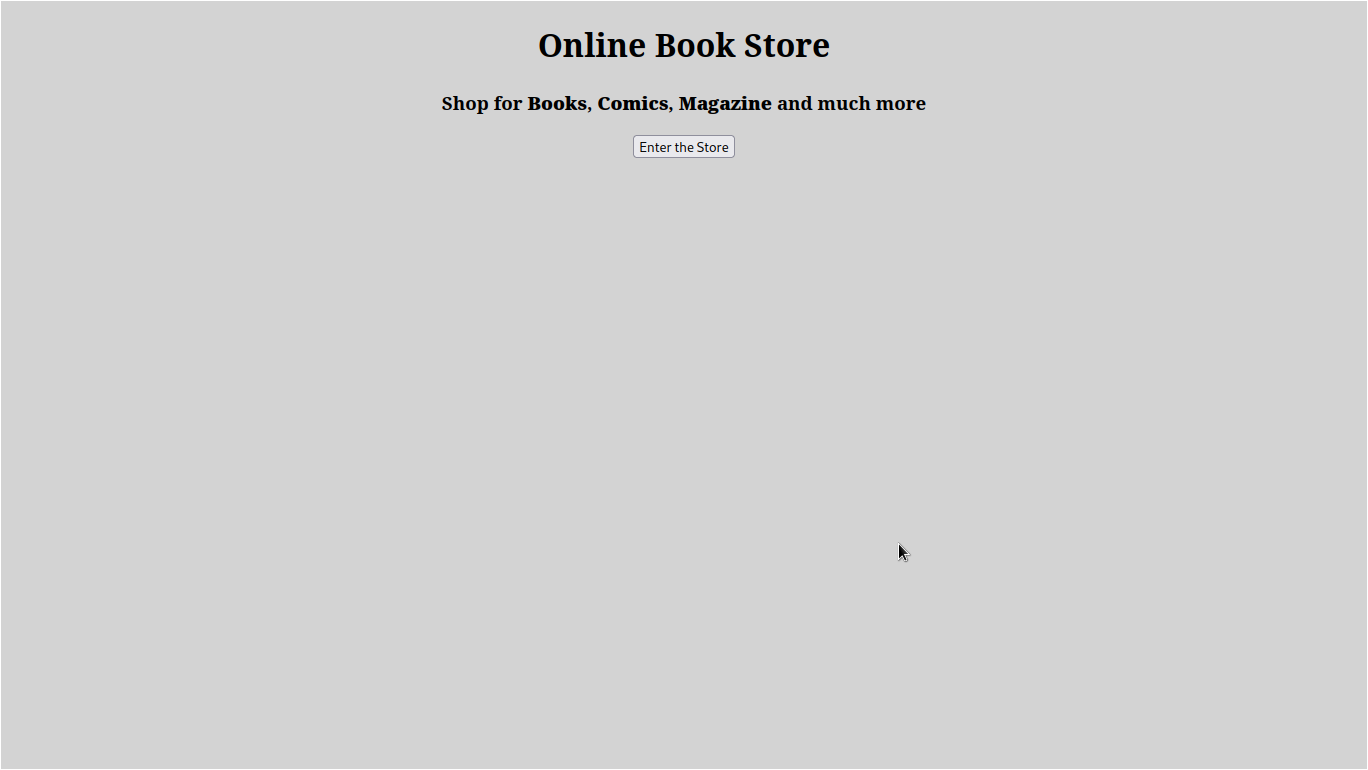
        &emsp;

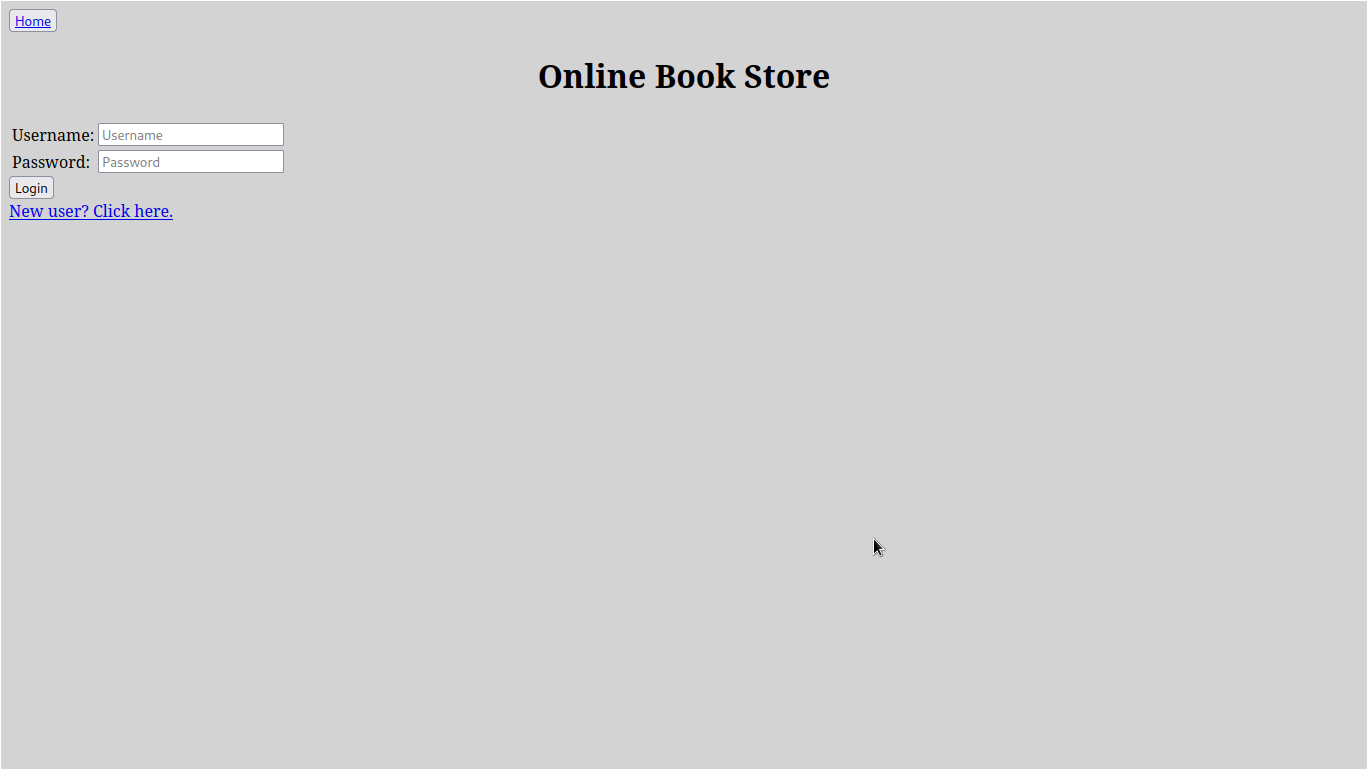
        <button type="button"><a href="ques7InventoryPage.html">Catalog</a></button>

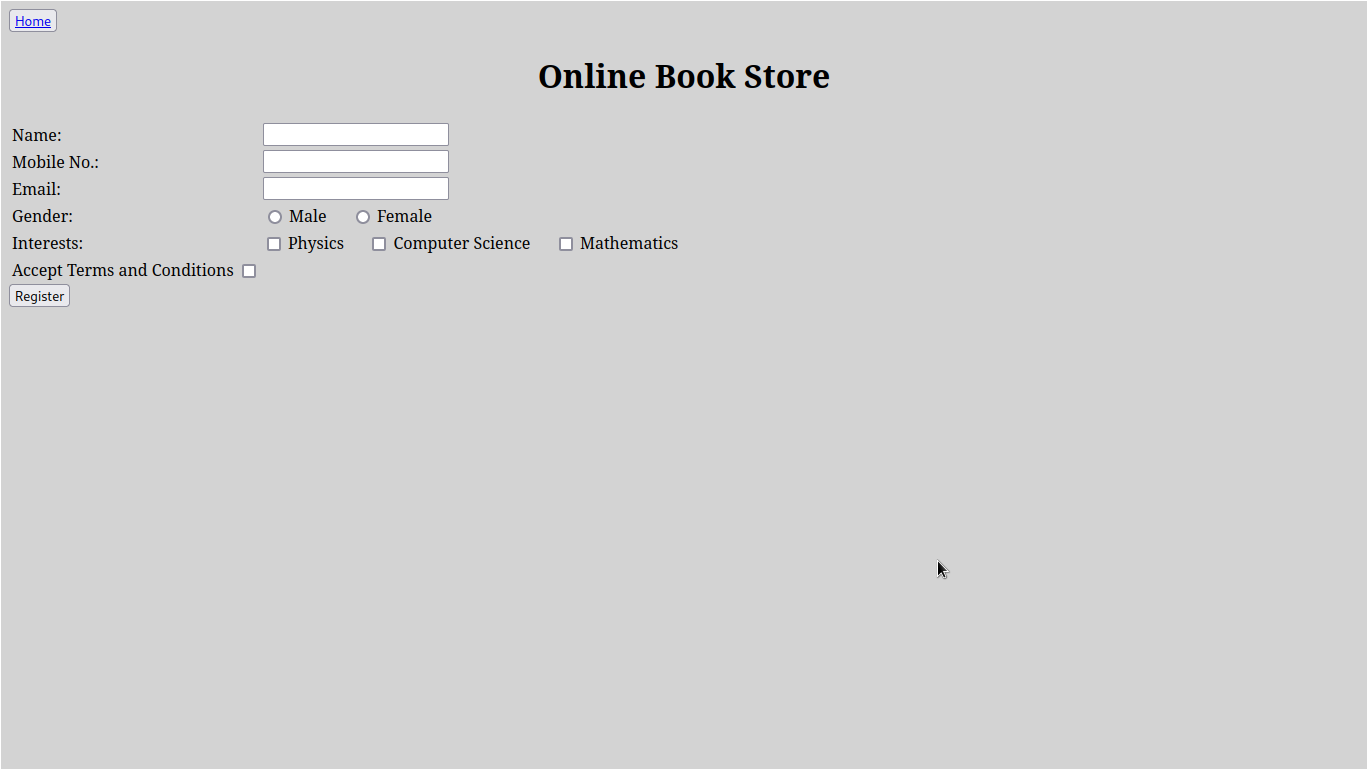
    </body>

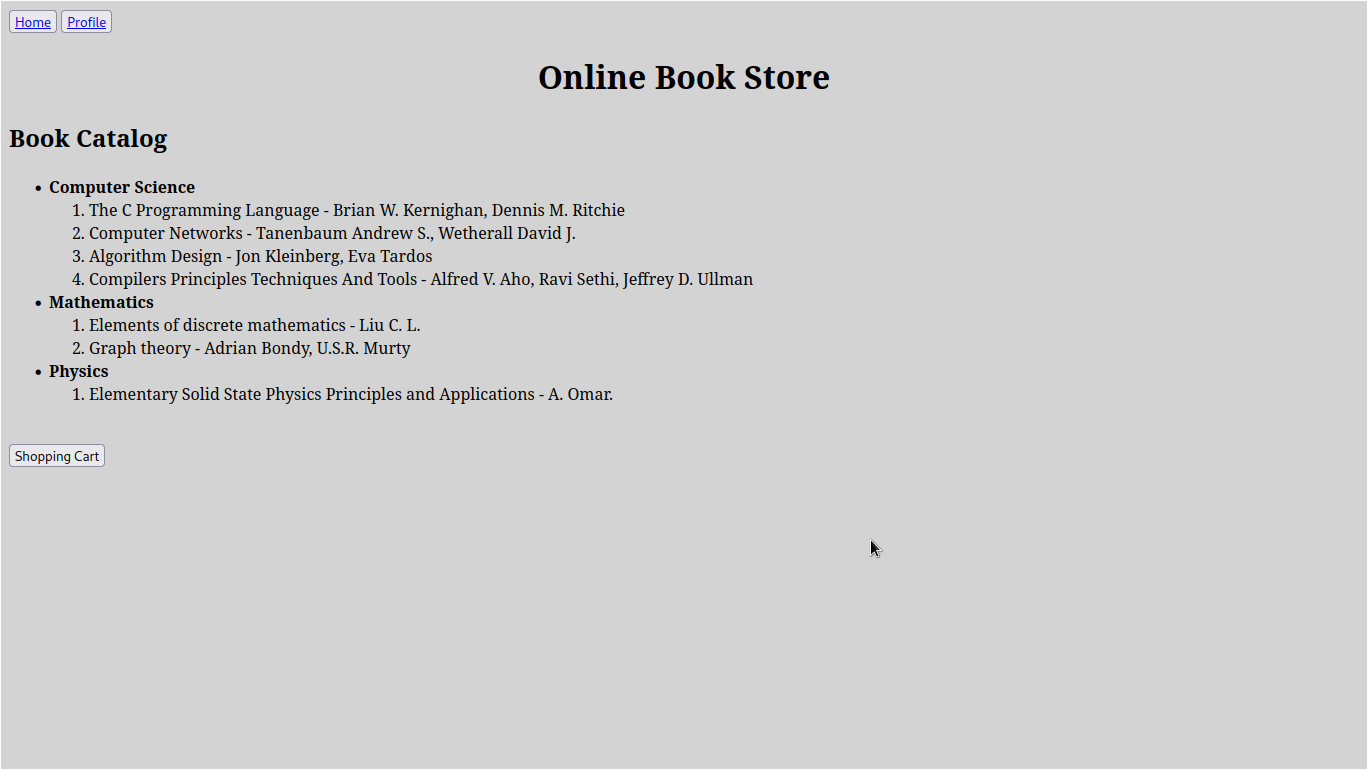
</html>

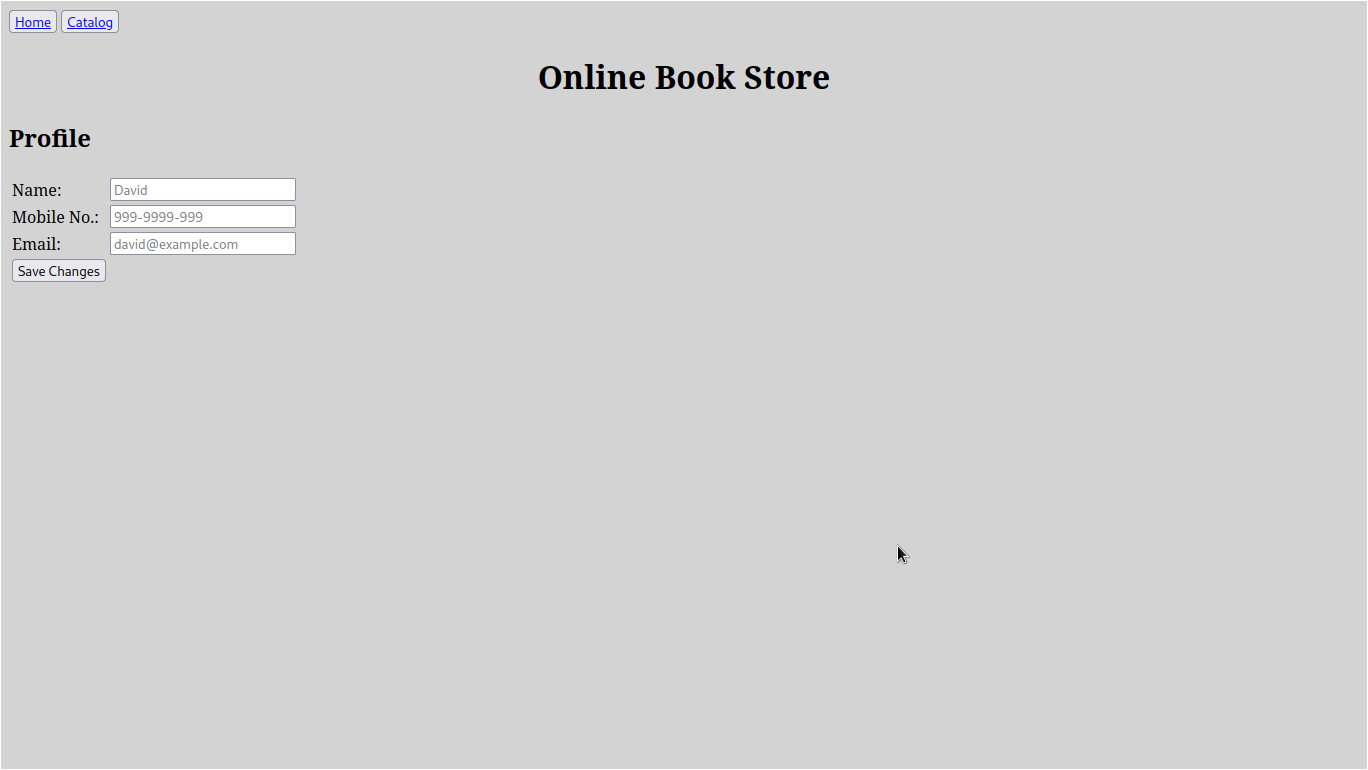
Output:

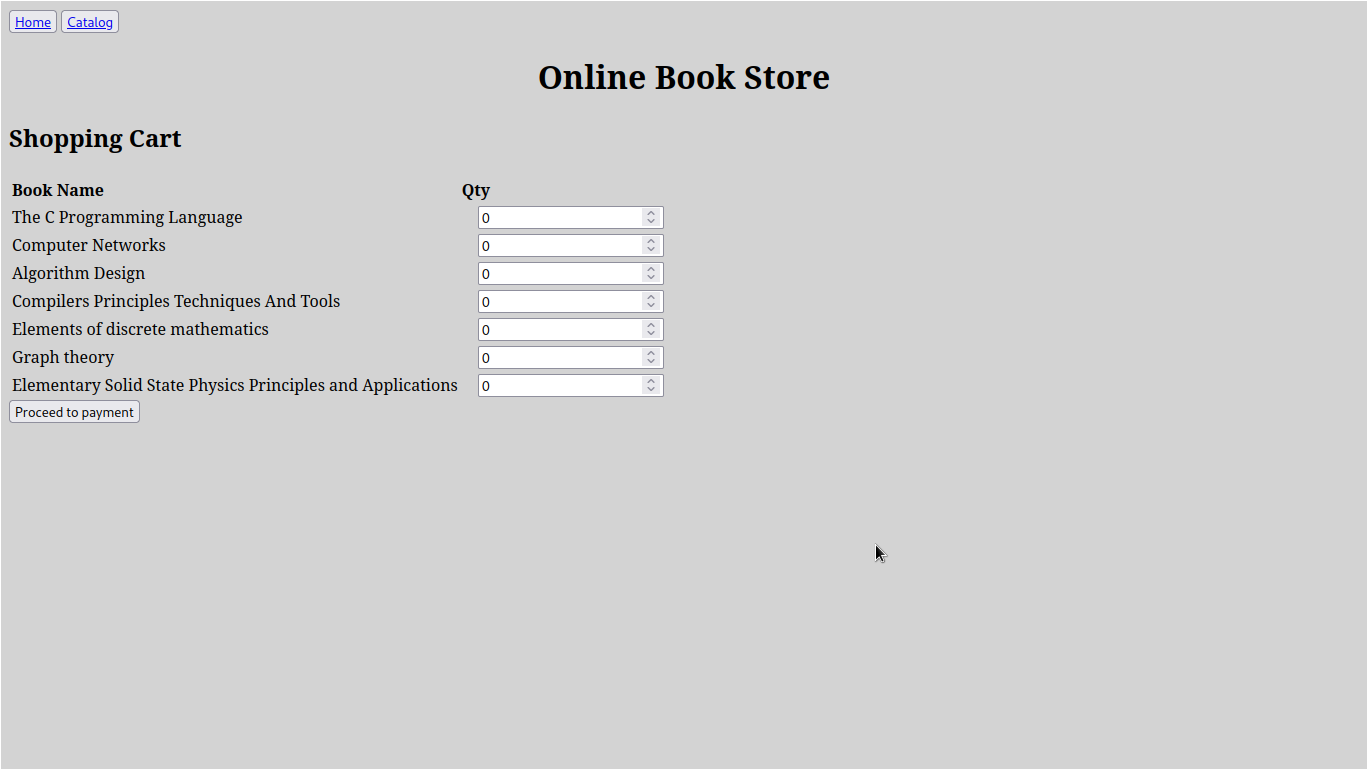
Home Page:  


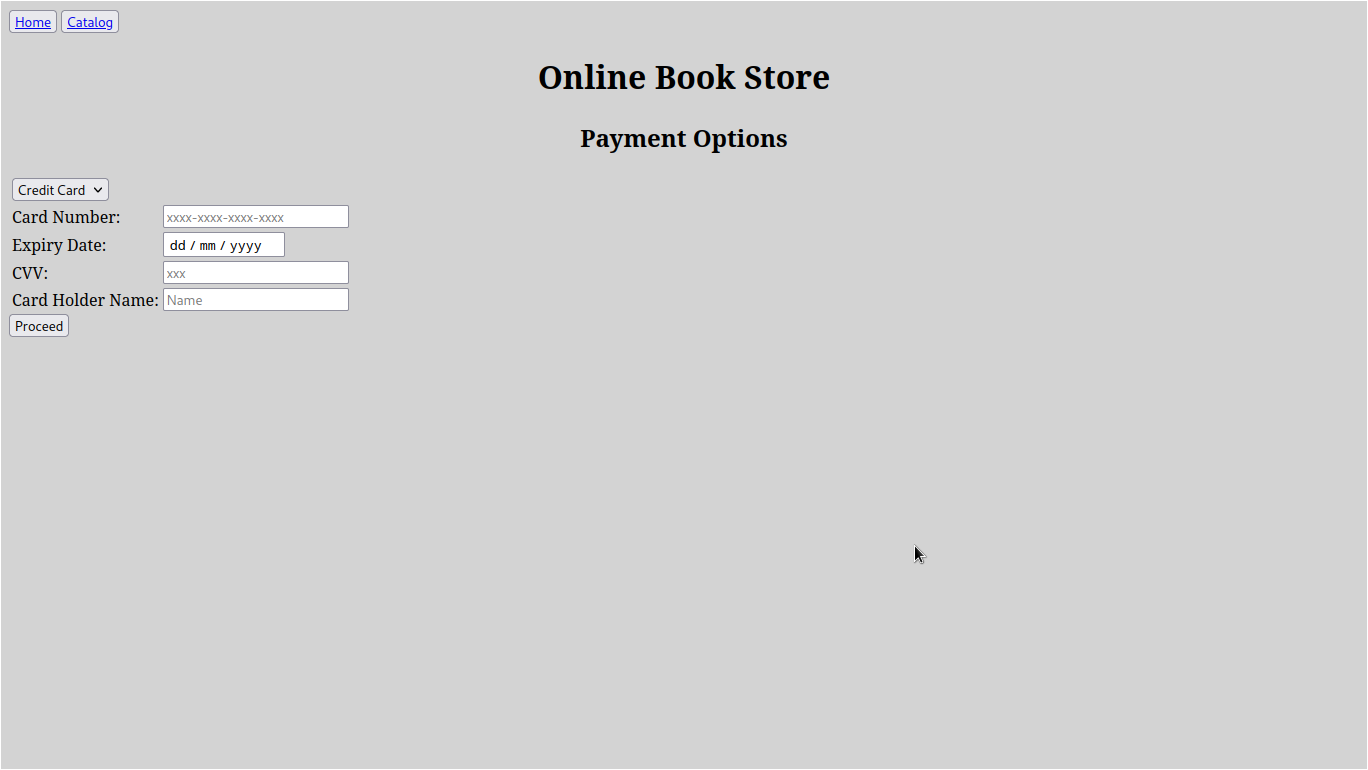
Login Page:  


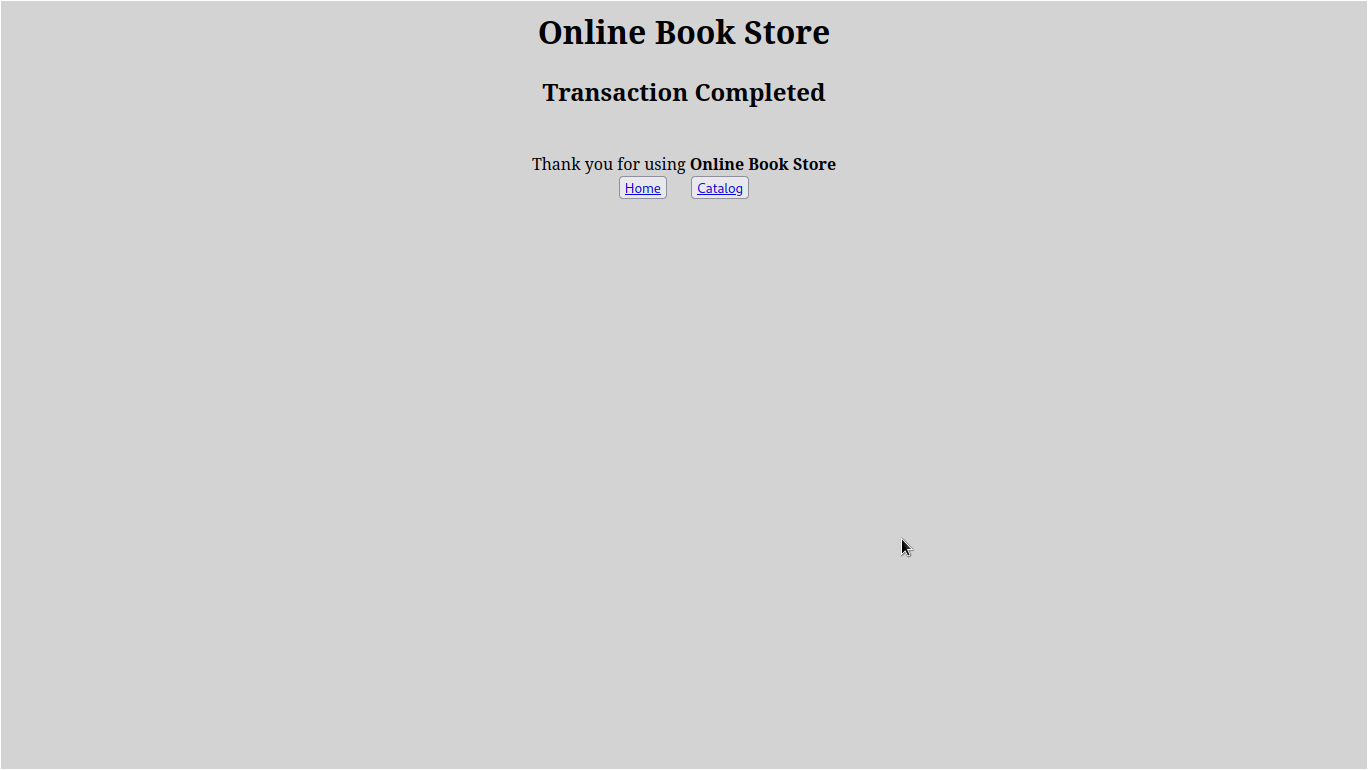
Register Page:  


Catalog Page:  


User Profile Page:  


Shopping Cart Page:  


Payment Page:  


Conformation Page:  


**NETWORK ALGORITHMS PRACTICAL**

Ques1: Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.

Solution:  
Code:

#include <iostream>

#include <string>

using namespace std;

char XOR(char a, char b){

    return a==b ? '0' : '1';

}

string CRC(string data, string gen\_fun){

    for(int i = 0; i < gen\_fun.length()-1; i++){

        data = data + '0';

    }

    while(data.length() >= gen\_fun.length()){

        for(int i = 0; i < gen\_fun.length(); i++){

            data[i] = XOR(data[i],gen\_fun[i]);

        }

        int index = -1;

        for(int i = 0; i < gen\_fun.length(); i++){

            if(data[i] == '1'){

                index = i;

                break;

            }

        }

        if(index == -1){

            data = data.substr(gen\_fun.length()-1);

        }

        else{

            data = data.substr(index);

        }

    }

    return data;

}

void sender(){

    string data, gen\_fun, temp;

    cout << "Enter the message : ";

    cin >> data;

    cout << "Enter the generating function : ";

    cin >> gen\_fun;

    temp = CRC(data,gen\_fun);

    for(int  i = 0; i < gen\_fun.length()-temp.length()-1; i++){

        data = data + '0';

    }

    data = data + temp;

    cout << "Message to send : " << data << "\n";

}

void receiver(){

    string data, gen\_fun;

    cout << "Enter the message received : ";

    cin >> data;

    cout << "Enter the generating function : ";

    cin >> gen\_fun;

    data = CRC(data,gen\_fun);

    int index = -1;

    for(int i = 0; i < data.length(); i++){

        if(data[i] == '1'){

            index = i;

            break;

        }

    }

    cout << "CRC: " << data << "\n";

    if(index == -1){

        cout << "Message with no error received\n";

    }

    else{

        cout << "Error detected in the message\n";

    }

}

int main(){

    char choice = ' ';

    while(choice != 'x'){

        cout << "1. Sender\n";

        cout << "2. Receiver\n";

        cout << "x. Exit(x)\n";

        cout << ">";

        cin >> choice;

        switch(choice){

            case '1':

                sender();

                break;

            case '2':

                receiver();

                break;

            case 'x':

                cout << "Exiting\n";

                break;

            default:

                cout << "Invalid!\n";

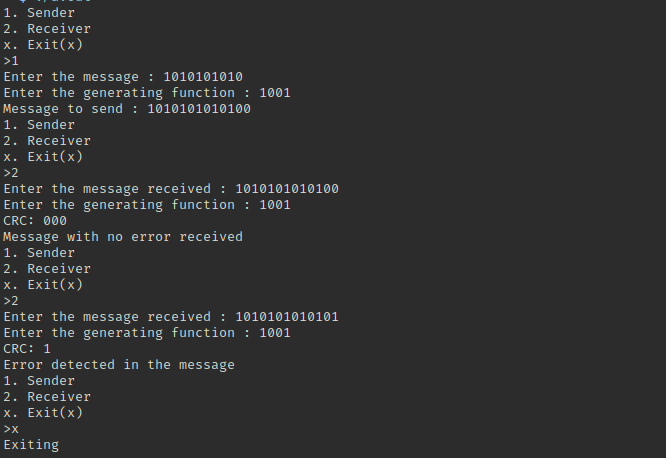
                break;

        }

    }

    return 0;

}

Output:  


Ques2: Simulate and implement stop and wait protocol for noisy channel.

Solution:  
Code:

#include <iostream>

#include <string>

#include <cstdlib>

#include <ctime>

#include <unistd.h>

using namespace std;

void sender(int random, int &i){

    switch(random){

        case 0:

            cout << "Sender: Acknowledgement " << i << " received\n";

            i++;

            break;

        case 1:

        case 2:

            cout << "Sender: Waiting for Acknowledgement\n";

            sleep(1);

            cout << "Sender: Time Out!\n";

            cout << "Sender: Resending the frame\n";

            break;

        case 3:

            cout << "Sender: Acknowledgement " << i-1 << " recevied\n";

            cout << "Sender: Wrong Acknowledgement\n";

            cout << "Resending the frame\n";

            break;

    }

}

void receiver(int random, int i){

    switch(random){

        case 0:

        case 1:

            cout << "Receiver: frame: " << i << " received\n";

            cout << "Receiver: Sending Acknowledgement " << i << " \n";

            break;

        case 2:

            sleep(1);

            cout << "(Receiver did not receive any frame)\n";

            break;

        case 3:

            cout << "Receiver: frame: " << i-1 << " received\n";

            cout << "Receiver: Sending Acknowledgement " << i-1 << "\n";

            break;

    }

}

int main(){

    int frames;

    int random;

    double ran;

    srand(time(NULL));

    cout << "Enter the number of frames: ";

    cin >> frames;

    for(int i = 0; i < frames; ){

        cout << "\n";

        ran = (rand()/double(RAND\_MAX));

        random = (ran < 0.25) ? 0:(ran < 0.5) ? 1:(ran < 0.75) ? 2:3;

        cout << "Sender: Sending frame: " << i << "\n";

        receiver(random,i);

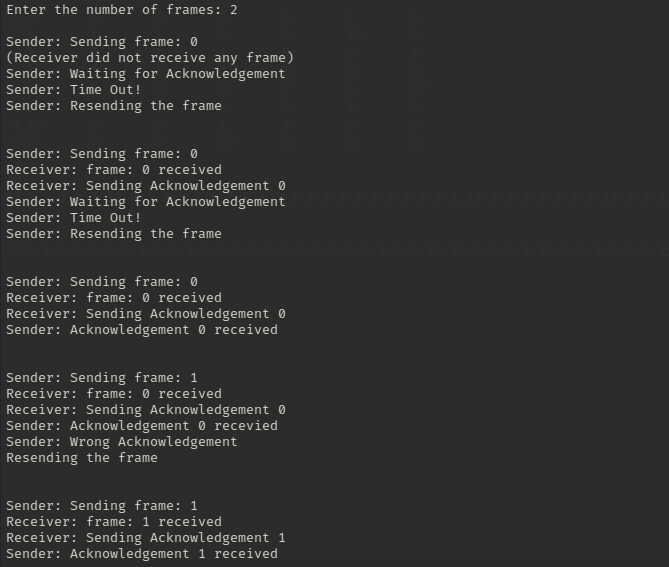
        sender(random,i);

        cout << "\n";

    }

    return 0;

}

Output:  


Ques3: Simulate and implement go back n sliding window protocol.

Solution:  
Code:  
#include <iostream>

#include <ctime>

#include <cstdlib>

using namespace std;

void sender(int i, int frames, int w\_size){

    int temp = i;

    for(i; i < temp+w\_size && i < frames; i++){

        cout << "Sender: Sending frame: " << i << "\n";

    }

    cout << "\n";

}

void receiver(int &i, int frames, int w\_size){

    srand(time(NULL));

    int temp = i;

    for(int j = i; j < temp+w\_size && j < frames; j++){

        double ran = rand()/double(RAND\_MAX);

        if(ran > 0.5){

            cout << "Receiver: frame: " << j << " received\n";

            i++;

        }

        else{

            cout << "Receiver: frame: " << j << " not received\n";

            break;

        }

    }

    cout << "\n";

}

void goBackN(int frames, int w\_size){

    int i = 0;

    while(i < frames){

        sender(i,frames,w\_size);

        receiver(i,frames,w\_size);

    }

}

int main(){

    int frames, w\_size;

    cout << "Enter the number of frames: ";

    cin >> frames;

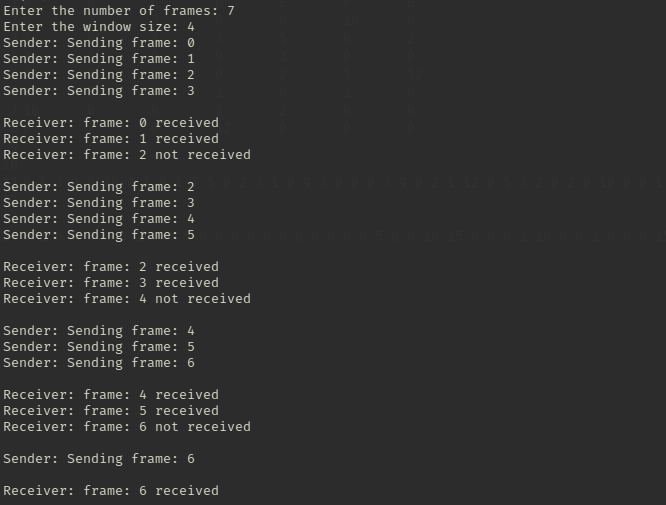
    cout << "Enter the window size: ";

    cin >> w\_size;

    goBackN(frames,w\_size);

    return 0;

}

Output:  


Ques4: Simulate and implement selective repeat sliding window protocol.

Solution:  
Code:

#include <iostream>

#include <vector>

#include <cstdlib>

#include <ctime>

#include <algorithm>

using namespace std;

void sender(vector<int> frame, int w\_size){

    for(int i = 0; i < w\_size && i < frame.size(); i++){

        cout << "Sender: sending frame: " << frame[frame.size()-i-1] << "\n";

    }

    cout << "\n";

}

vector<int> receiver(vector<int> frame, int w\_size){

    srand(time(NULL));

    vector<int> temp;

    for(int i = 0; i < w\_size && (!frame.empty()); i++){

        double ran = rand()/double(RAND\_MAX);

        if(ran > 0.5){

            cout << "Receiver: frame: " << frame[frame.size()-1] << " received\n";

            frame.pop\_back();

        }

        else{

            cout << "Receiver: frame: " << frame[frame.size()-1] << " not received\n";

            temp.push\_back(frame[frame.size()-1]);

            frame.pop\_back();

        }

    }

    while(!temp.empty()){

        frame.push\_back(temp[temp.size()-1]);

        temp.pop\_back();

    }

    cout << "\n";

    return frame;

}

void selectiveRepeat(int frames, int w\_size){

    vector<int> frame;

    for(int i = 0; i < frames; i++){

        frame.push\_back(i);

    }

    reverse(frame.begin(), frame.end());

    while(!frame.empty()){

        sender(frame,w\_size);

        frame=receiver(frame,w\_size);

    }

}

int main(){

    int frames, w\_size;

    cout << "Enter the number of frames: ";

    cin >> frames;

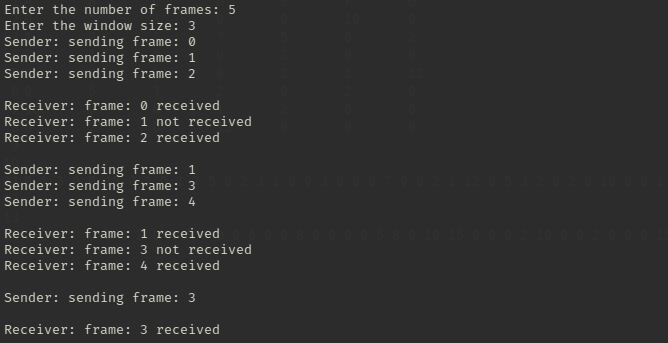
    cout << "Enter the window size: ";

    cin >> w\_size;

    selectiveRepeat(frames,w\_size);

    return 0;

}

Output:  


Ques5: Shortest Path algorithm.

Solution:  
Code:

#include <iostream>

#include <vector>

#include <limits>

#include <algorithm>

using namespace std;

void shortestPath(vector<vector<int>> graph, int source, int nodes, vector<int> &dist, vector<int> &path){

    vector<int> unvisited, visited, layer;

    int size;

    for(int i = 0; i < nodes; i++){

        dist.push\_back(numeric\_limits<int>::max());

        path.push\_back(numeric\_limits<int>::max());

        if(i != source-1){

            unvisited.push\_back(i);

        }

    }

    dist[source-1] = 0;

    path[source-1] = source-1;

    visited.push\_back(source-1);

    while(!visited.empty()){

        size=unvisited.size();

        for(int i = 0; i < visited.size(); i++){

            for(int j = 0; j < nodes; j++){

                if(graph[visited[i]][j] != numeric\_limits<int>::max()){

                    if(find(unvisited.begin(), unvisited.end(), j) != unvisited.end()){

                        layer.push\_back(j);

                        remove(unvisited.begin(), unvisited.end()+1, j);

                        unvisited.resize(unvisited.size()-1);

                        size--;

                    }

                    /\*if(path[j] == numeric\_limits<int>::max()){

                        dist[j] = graph[visited[i]][j] + dist[visited[i]];

                        path[j] = visited[i];

                    }\*/

                    if(dist[j] > dist[visited[i]] + graph[visited[i]][j]){

                        dist[j] = dist[visited[i]] + graph[visited[i]][j];

                        path[j] = visited[i];

                    }

                }

            }

        }

        visited=layer;

        layer.clear();

    }

}

void printPath(vector<int> dist, vector<int> path, int nodes, int source){

    for(int i = 0; i < nodes; i++){

        cout << "Node " << i+1 << ": " << i+1 << "<-";

        int j = i;

        while(path[j] != source-1){

            cout << path[j]+1 << "<-";

            j = path[j];

        }

        cout << source;

        cout << " Distance: " << dist[i] << "\n";

    }

}

int main(){

    vector<vector<int>> graph;

    vector<int> ele, distance, path;

    int d, nodes;

    cout << "Enter the number of Nodes: ";

    cin >> nodes;

    for(int i = 0; i < nodes; i++){

        cout << "Enter the distance from node " << i+1 << " to all the Nodes: ";

        for(int j = 0; j < nodes; j++){

            cin >> d;

            if(d == 0){

                ele.push\_back(numeric\_limits<int>::max());

            }

            else{

                ele.push\_back(d);

            }

        }

        graph.push\_back(ele);

        ele.clear();

    }

    cout << "Enter the Source Node: ";

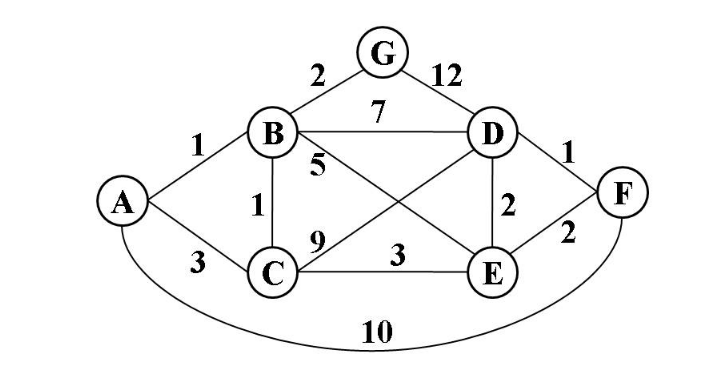
    cin >> d;

    shortestPath(graph,d,nodes,distance,path);

    printPath(distance,path,nodes,d);

    return 0;

}

Graph Used:  


Output:  
