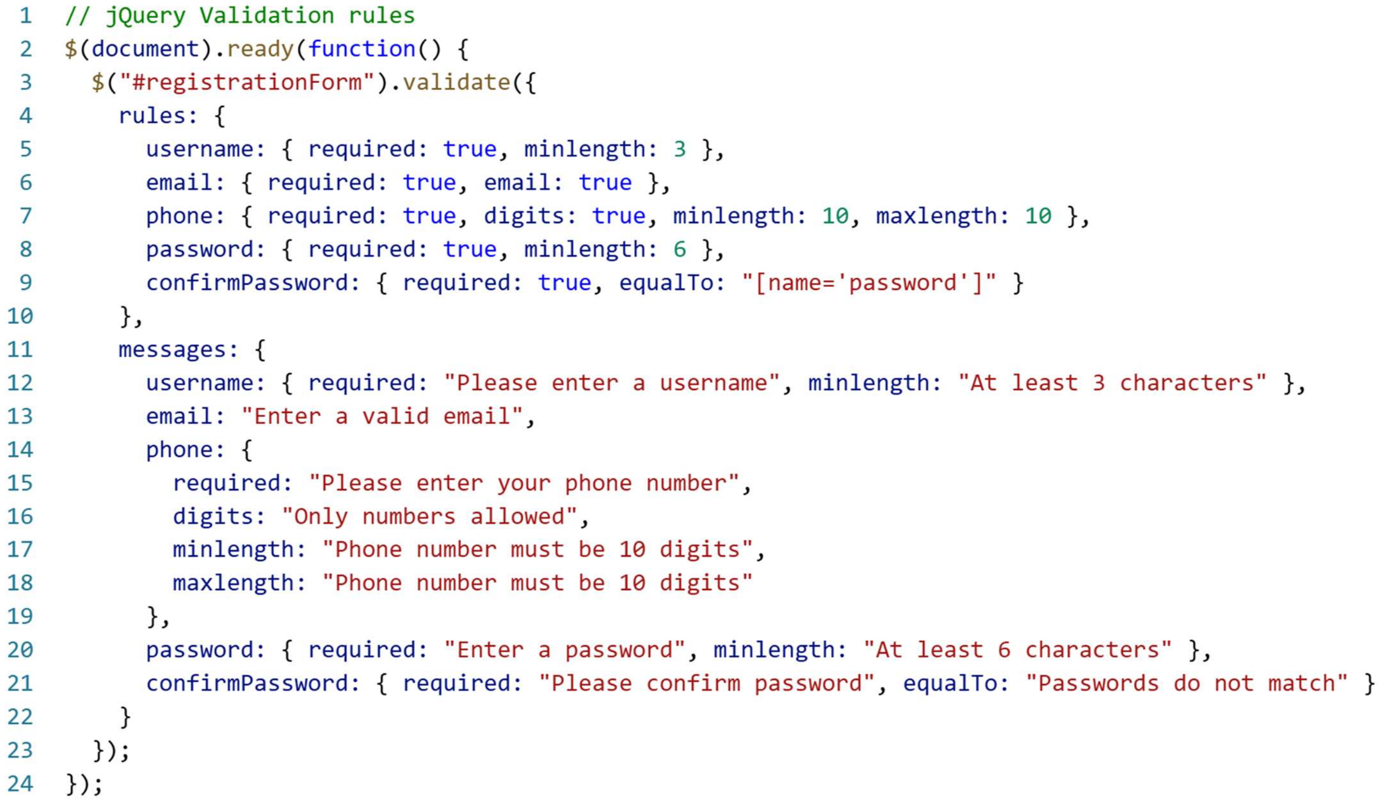
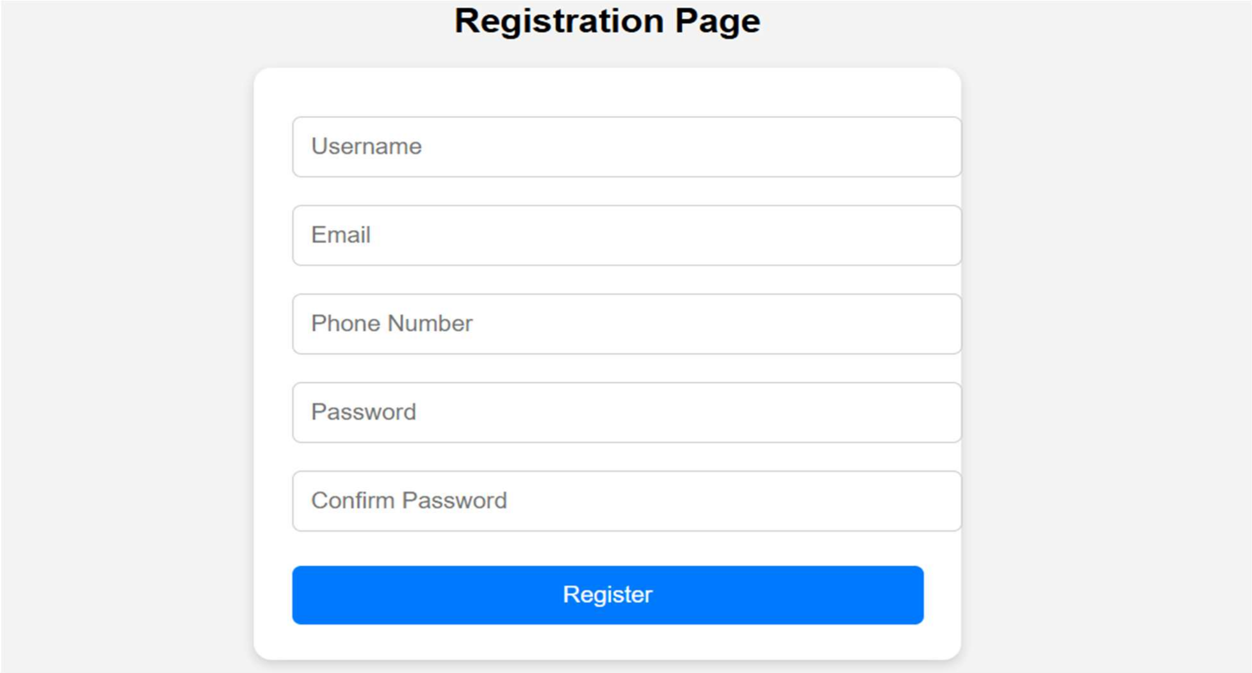
Experiment 4.1:

USER REGISTRATION FORM AND PERFORM INPUT VALIDATION

CODE:



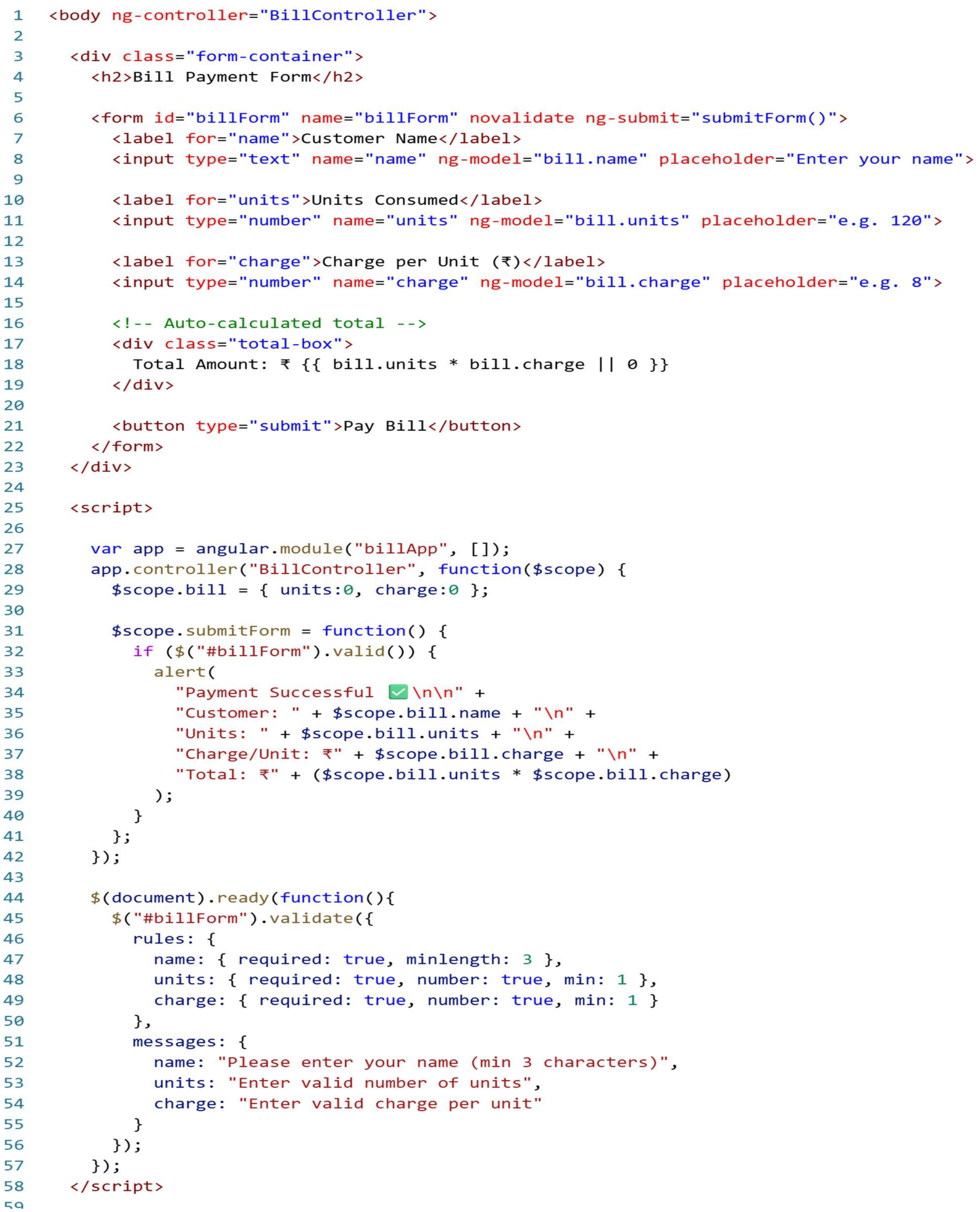
```
// jQuery Validation rules
$(document).ready(function() {
  $("#registrationForm").validate({
    rules: {
      username: { required: true, minlength: 3 },
      email: { required: true, email: true },
      phone: { required: true, digits: true, minlength: 10, maxlength: 10 },
      password: { required: true, minlength: 6 },
      confirmPassword: { required: true, equalTo: "[name='password']" }
    },
    messages: {
      username: { required: "Please enter a username", minlength: "At least 3 characters" },
      email: "Enter a valid email",
      phone: {
        required: "Please enter your phone number",
        digits: "Only numbers allowed",
        minlength: "Phone number must be 10 digits",
        maxlength: "Phone number must be 10 digits"
      },
      password: { required: "Enter a password", minlength: "At least 6 characters" },
      confirmPassword: { required: "Please confirm password", equalTo: "Passwords do not match" }
    }
  });
});
```



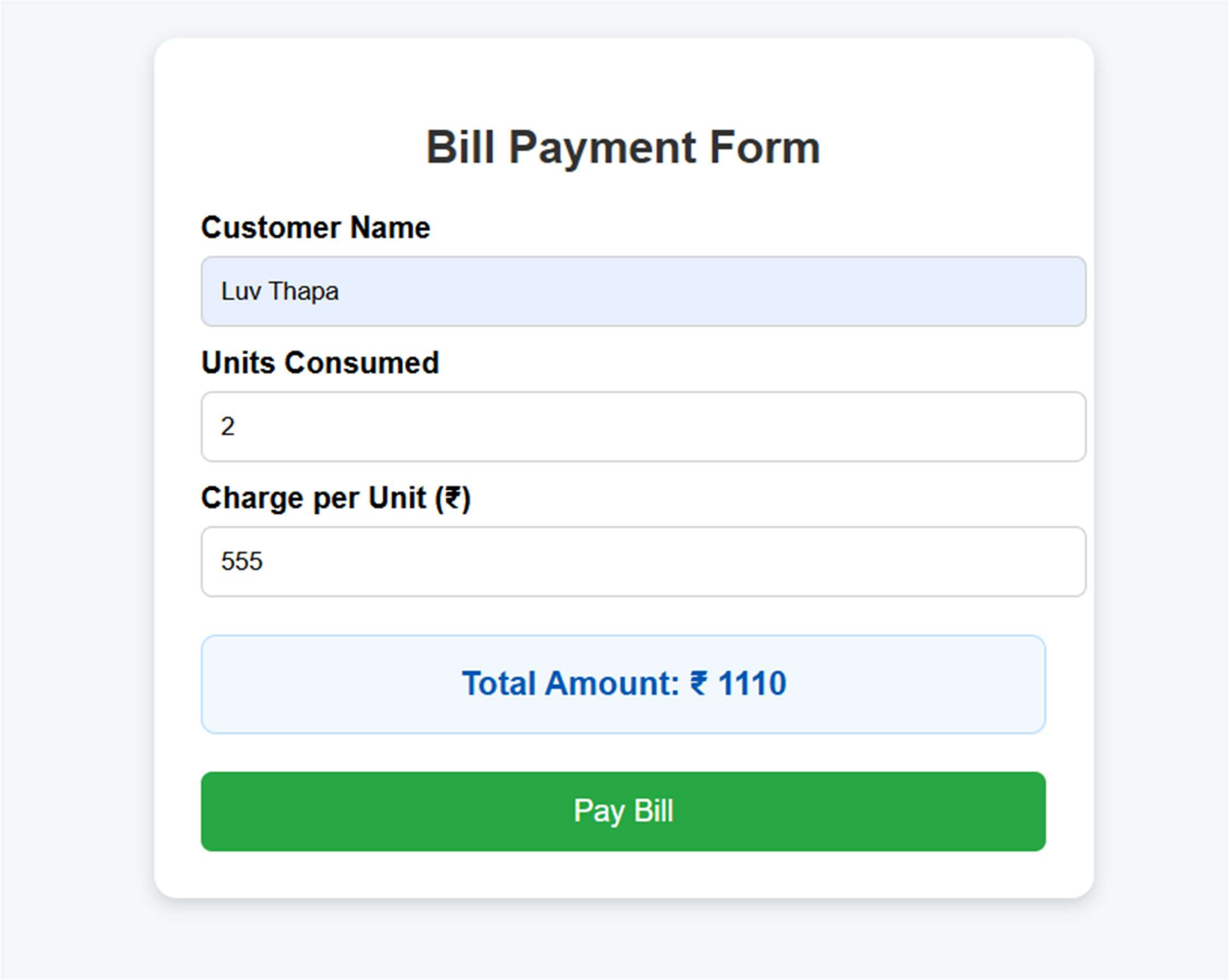
Experiment 4.2:

APPLICATION FOR BILL PAYMENT RECORD

CODE:



```
<body ng-controller="BillController">

  <div class="form-container">
    <h2>Bill Payment Form</h2>

    <form id="billForm" name="billForm" novalidate ng-submit="submitForm()">
      <label for="name">Customer Name</label>
      <input type="text" name="name" ng-model="bill.name" placeholder="Enter your name">

      <label for="units">Units Consumed</label>
      <input type="number" name="units" ng-model="bill.units" placeholder="e.g. 120">

      <label for="charge">Charge per Unit (₹)</label>
      <input type="number" name="charge" ng-model="bill.charge" placeholder="e.g. 8">

      <!-- Auto-calculated total -->
      <div class="total-box">
        Total Amount: ₹ {{ bill.units * bill.charge || 0 }}
      </div>

      <button type="submit">Pay Bill</button>
    </form>
  </div>

  <script>

    var app = angular.module("billApp", []);
    app.controller("BillController", function($scope) {
      $scope.bill = { units:0, charge:0 };

      $scope.submitForm = function() {
        if ($("#billForm").valid()) {
          alert(
            "Payment Successful ✅\n\n" +
            "Customer: " + $scope.bill.name + "\n" +
            "Units: " + $scope.bill.units + "\n" +
            "Charge/Unit: ₹" + $scope.bill.charge + "\n" +
            "Total: ₹" + ($scope.bill.units * $scope.bill.charge)
          );
        }
      };
    });

    $(document).ready(function(){
      $("#billForm").validate({
        rules: {
          name: { required: true, minlength: 3 },
          units: { required: true, number: true, min: 1 },
          charge: { required: true, number: true, min: 1 }
        },
        messages: {
          name: "Please enter your name (min 3 characters)",
          units: "Enter valid number of units",
          charge: "Enter valid charge per unit"
        }
      });
    });
  </script>
```



Bill Payment Form

Customer Name

Luv Thapa

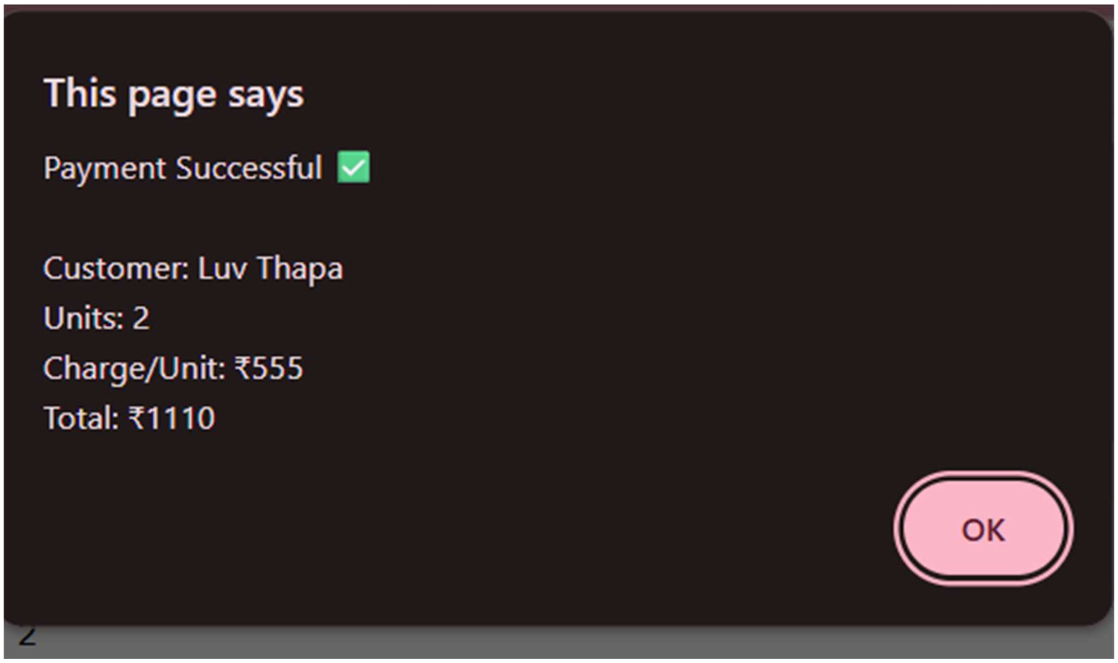
Units Consumed

2

Charge per Unit (₹)

555

Total Amount: ₹ 1110



This page says

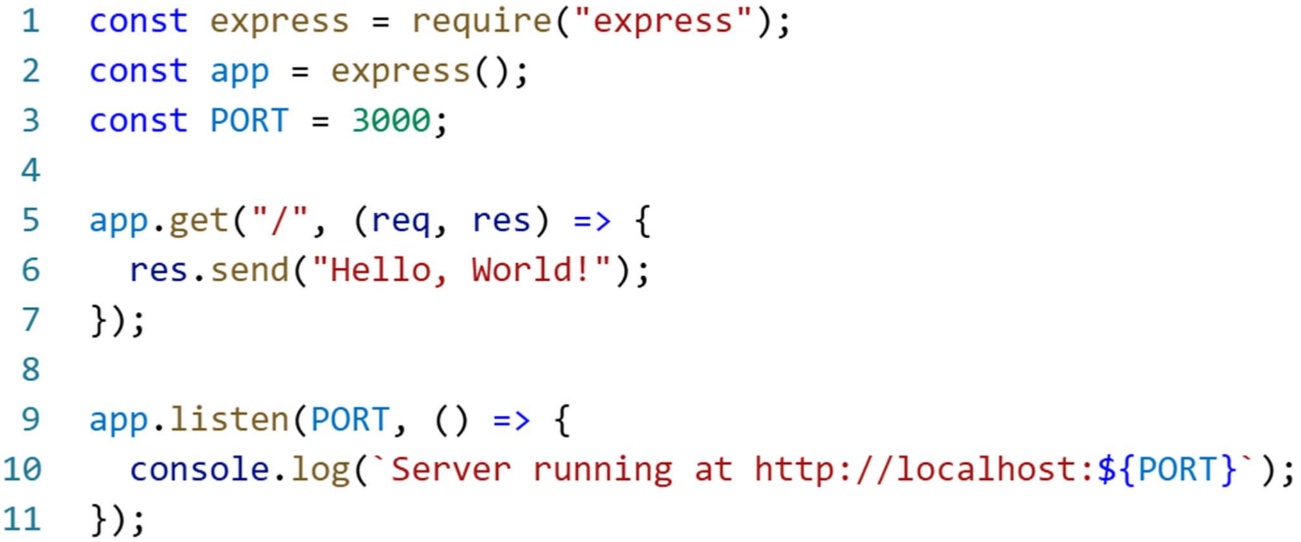
Payment Successful ✅

Customer: Luv Thapa
Units: 2
Charge/Unit: ₹555
Total: ₹1110

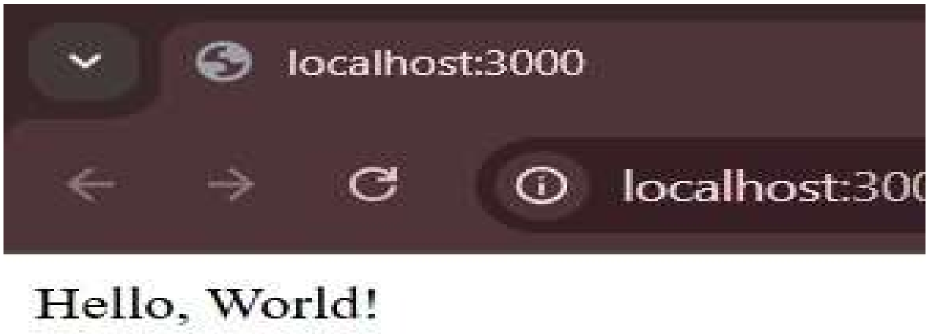
Experiment 5.1:

CREATE A SIMPLE “HELLO, WORLD!” SERVER

CODE:



```
const express = require("express");
const app = express();
const PORT = 3000;

app.get("/", (req, res) => {
  res.send("Hello, World!");
});

app.listen(PORT, () => {
  console.log(`Server running at http://localhost:${PORT}`);
});
```

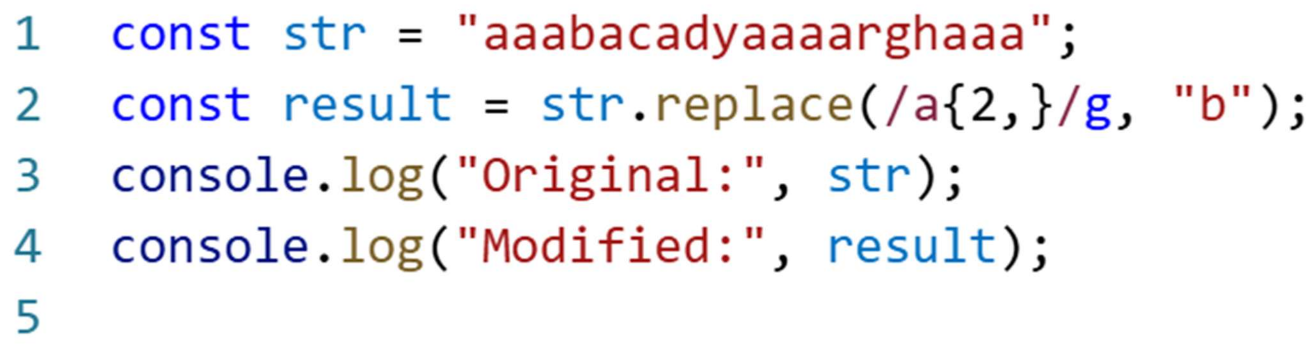


Hello, World!

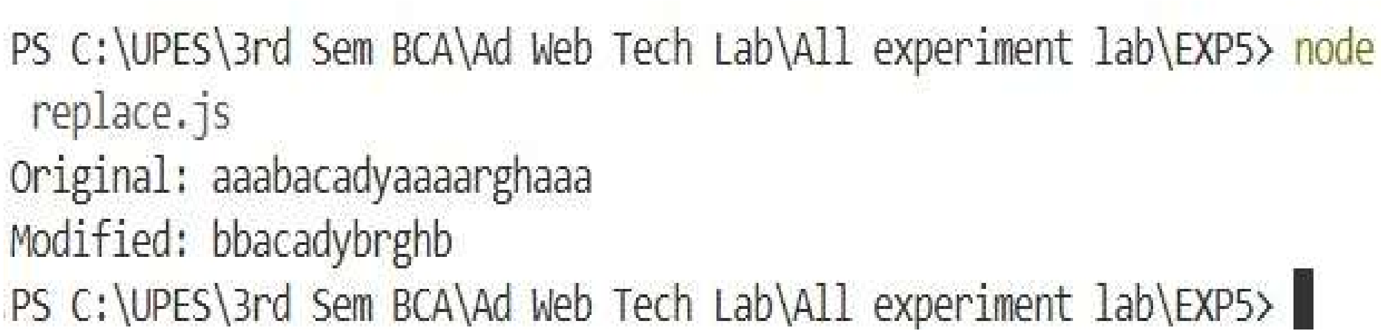
Experiment 5.2:

REPLACE TWO OR MORE A'S WITH THE LETTER B

CODE:



```
const str = "aaabacadyaaaarghaaa";
const result = str.replace(/a{2,}/g, "b");
console.log("Original:", str);
console.log("Modified:", result);

```

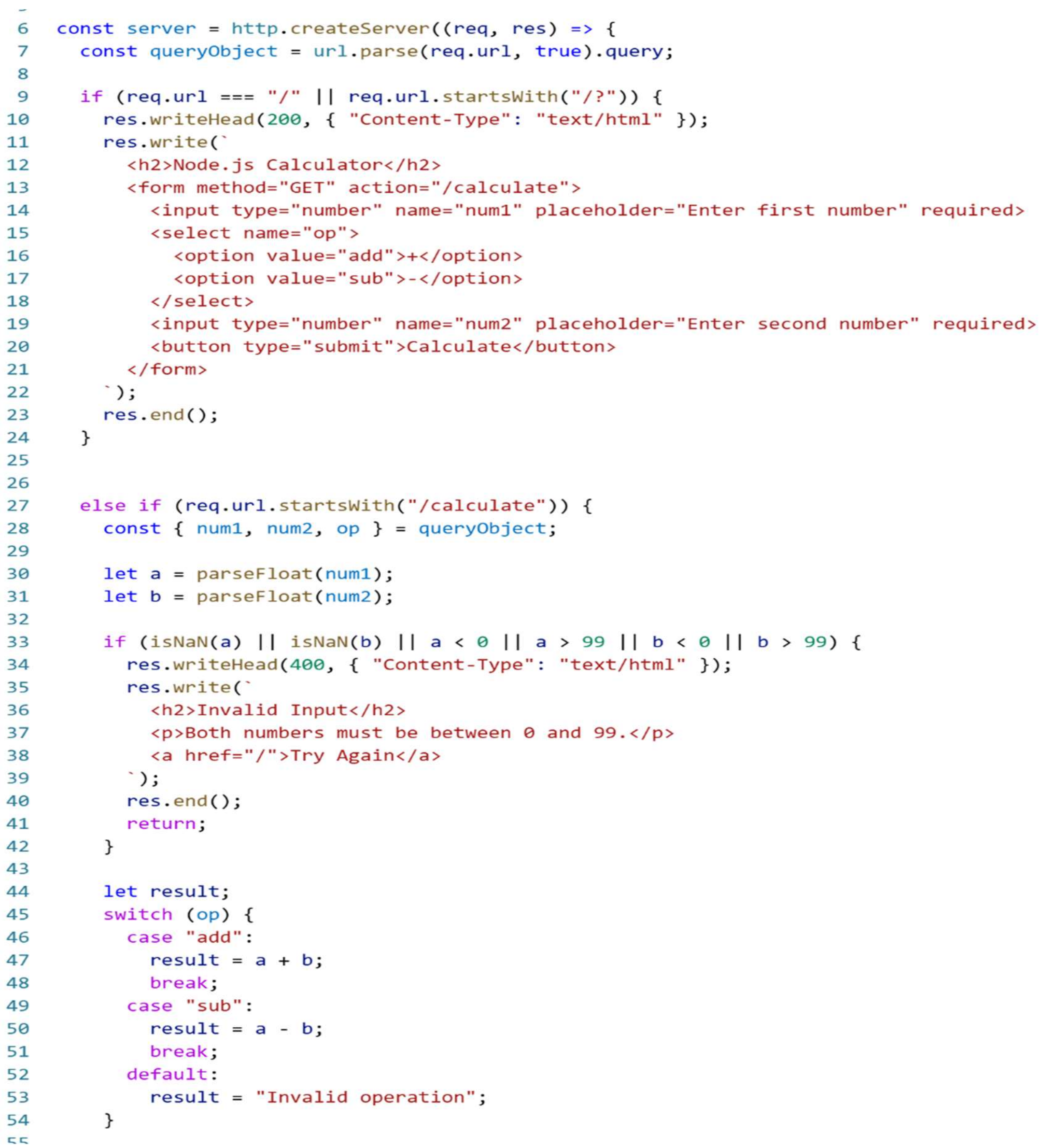


```
PS C:\UPES\3rd Sem BCA\Ad Web Tech Lab\All experiment lab\EXP5> node
 replace.js
Original: aaabacadyaaaarghaaa
Modified: bbacadybrghb
PS C:\UPES\3rd Sem BCA\Ad Web Tech Lab\All experiment lab\EXP5> 
```

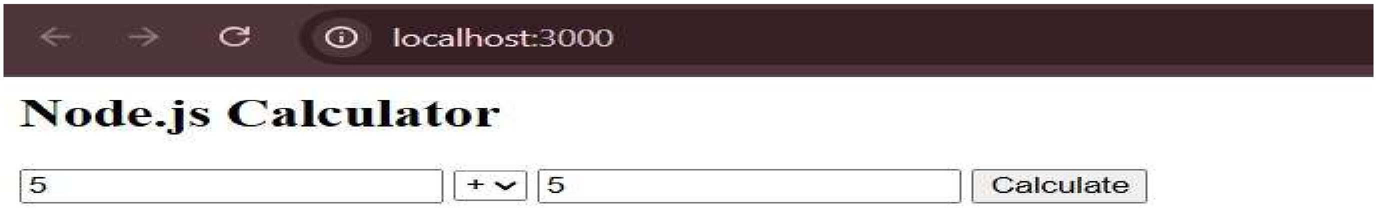
Experiment 6.1:

BASIC CALCULATOR THAT CAN PERFORM ARITHMETIC OPERATIONS

CODE:



```
const server = http.createServer((req, res) => {
  const queryObject = url.parse(req.url, true).query;

  if (req.url === "/" || req.url.startsWith("/?")) {
    res.writeHead(200, { "Content-Type": "text/html" });
    res.write(`
      <h2>Node.js Calculator</h2>
      <form method="GET" action="/calculate">
        <input type="number" name="num1" placeholder="Enter first number" required>
        <select name="op">
          <option value="add">+</option>
          <option value="sub">-</option>
        </select>
        <input type="number" name="num2" placeholder="Enter second number" required>
        <button type="submit">Calculate</button>
      </form>
    `);
    res.end();
  }


  else if (req.url.startsWith("/calculate")) {
    const { num1, num2, op } = queryObject;

    let a = parseFloat(num1);
    let b = parseFloat(num2);

    if (isNaN(a) || isNaN(b) || a < 0 || a > 99 || b < 0 || b > 99) {
      res.writeHead(400, { "Content-Type": "text/html" });
      res.write(`
        <h2>Invalid Input</h2>
        <p>Both numbers must be between 0 and 99.</p>
        <a href="/">Try Again</a>
      `);
      res.end();
      return;
    }

    let result;
    switch (op) {
      case "add":
        result = a + b;
        break;
      case "sub":
        result = a - b;
        break;
      default:
        result = "Invalid operation";
    }
```



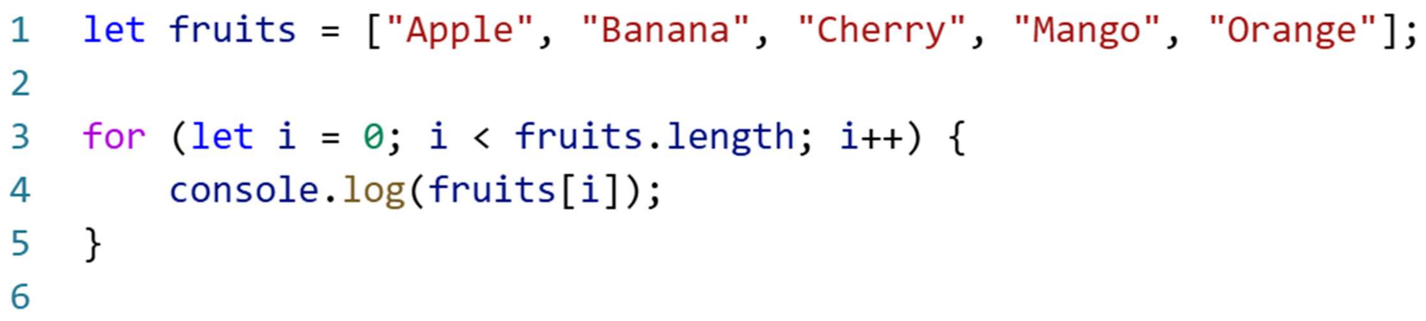
Node.js Calculator

5 + 5 Calculate

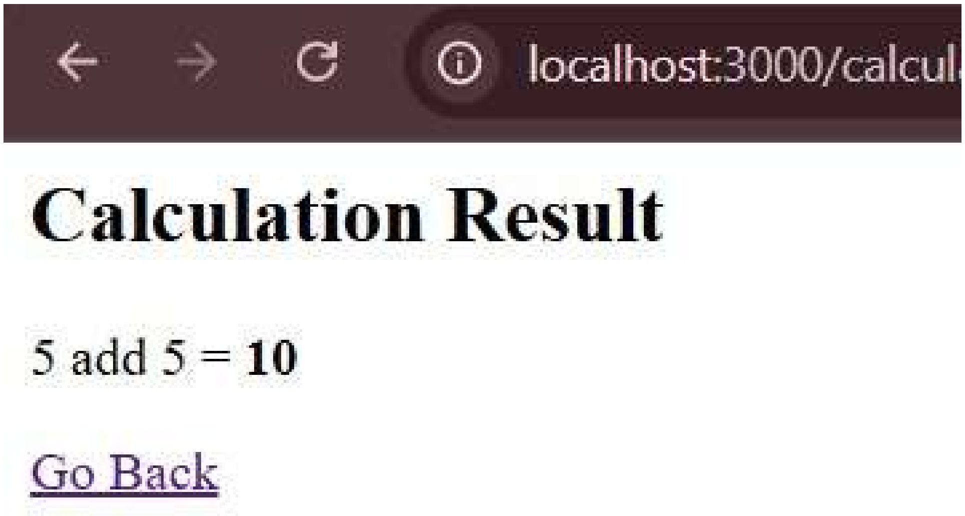
Experiment 6.2:

ITERATE OVER THE GIVEN ARRAY

CODE:



```
let fruits = ["Apple", "Banana", "Cherry", "Mango", "Orange"];

for (let i = 0; i < fruits.length; i++) {
    console.log(fruits[i]);
}

```



Calculation Result

5 add 5 = **10**

Go Back