

## Task-1:

**Create a student registration form with HTML5 validation for name, email, and age inputs:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>Student Registration Form</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <style>
    body { font-family: sans-serif; background: #f4f4fa; }
    form { max-width: 350px; margin: 2rem auto; background: #fff; padding: 1.5rem;
border-radius: 8px; box-shadow: 0 2px 8px #0001; }
    label, input { display: block; width: 100%; margin-bottom: 12px; }
    input[type="submit"] { background: #3560f6; color: #fff; border: none; border-radius:
4px; padding: 10px; cursor: pointer; }
    input[type="submit"]:hover { background: #233bb2; }
  </style>
</head>
<body>
  <form>
    <h2>Student Registration</h2>
    <label>Name
      <input type="text" name="name" required pattern="[A-Za-z ]{3,}" title="Enter at least 3
alphabetic characters"/>
    </label>
    <label>Email
      <input type="email" name="email" required />
    </label>
    <label>Age
      <input type="number" name="age" required min="16" max="120" />
    </label>
    <input type="submit" value="Register"/>
  </form>
</body>
</html>
```

## Task-2:

### Build a responsive banking UI with deposit/withdrawal buttons and balance display:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>Bank UI</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <style>
    body { font-family: sans-serif; background: #eaeef1; }
    .bank-container {
      max-width: 400px;
      background: #fff;
      margin: 3rem auto;
      border-radius: 10px;
      box-shadow: 0 2px 10px #0002;
      padding: 2rem;
      text-align: center;
    }
    .balance { font-size: 2rem; margin: 1rem 0; color: #3560f6; }
    .btns input[type="number"] { width: 100px; margin-right: 1rem; }
    .deposit, .withdraw {
      background: #3560f6;
      color: #fff; border: none; border-radius: 5px; padding: 8px 14px;
      margin: 0.5rem;
      cursor: pointer;
    }
    .withdraw { background: #f66c4f; }
    @media (max-width: 520px) {
      .bank-container { padding: 1rem; }
      .btns { display: block; }
      .btns > * { margin: 0.25rem auto; }
    }
  </style>
</head>
<body>
  <div class="bank-container">
```

```

<h2>Your Account</h2>
<div class="balance" id="balance">$0.00</div>
<div class="btns">
  <input type="number" id="amount" placeholder="Amount" min="1"/>
  <button class="deposit" onclick="deposit()">Deposit</button>
  <button class="withdraw" onclick="withdraw()">Withdraw</button>
</div>
<div id="message" style="margin-top: 1rem; color: #d64f3a;"></div>
</div>
<script>
let balance = 0;
function updateBalance() {
  document.getElementById('balance').innerText = `$$${balance.toFixed(2)}`;
}
function showMsg(msg, color) {
  const msgDiv = document.getElementById('message');
  msgDiv.style.color = color;
  msgDiv.innerText = msg;
  setTimeout(()=>msgDiv.innerText="",2000);
}
function deposit() {
  let amt = Number(document.getElementById('amount').value);
  if(amt > 0) {
    balance += amt;
    updateBalance();
    showMsg('Deposit Successful', '#36ad6e');
  } else {
    showMsg('Enter a valid amount.', '#d64f3a');
  }
  document.getElementById('amount').value = "";
}
function withdraw() {
  let amt = Number(document.getElementById('amount').value);
  if(amt > 0 && amt <= balance) {
    balance -= amt;
    updateBalance();
    showMsg('Withdrawal Successful', '#36ad6e');
  } else if(amt > balance) {
    showMsg('Insufficient funds.', '#d64f3a');
  }
}

```

```
    } else {  
        showMsg('Enter a valid amount.', '#d64f3a');  
    }  
    document.getElementById('amount').value = "";  
}  
updateBalance();  
</script>  
</body>  
</html>
```

## Task-3:

### Admin Dashboard with CSS Grid & Dynamic Theme Switching:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>Admin Dashboard</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <style>
    :root {
      --bg: #f4f4fa;
      --sidebar: #222b45;
      --sidebar-text: #fff;
      --main: #fff;
      --accent: #3560f6;
      --text: #1c1e21;
    }
    [data-theme="dark"] {
      --bg: #181c23;
      --sidebar: #233;
      --sidebar-text: #c8cfff;
      --main: #24273a;
      --accent: #e9b96e;
      --text: #f5f7fa;
    }
    body {
      font-family: 'Segoe UI', sans-serif;
      margin: 0; background: var(--bg); color: var(--text);
      min-height: 100vh;
      transition: background 0.3s, color 0.3s;
    }
    .dashboard {
      min-height: 100vh; display: grid;
      grid-template-areas:
        "sidebar header"
        "sidebar main";
      grid-template-columns: 220px 1fr;
      grid-template-rows: 60px 1fr;
```

```
}
.sidebar {
  grid-area: sidebar;
  background: var(--sidebar);
  color: var(--sidebar-text);
  padding: 1.5rem 1rem;
  min-height: 0;
  display: flex;
  flex-direction: column;
  gap: 1.5rem;
}
.sidebar h2 { color: var(--accent); font-size: 1.4rem; }
.sidebar a {
  color: var(--sidebar-text); text-decoration: none;
  padding: 0.4rem 0.8rem; border-radius: 4px;
}
.sidebar a.active, .sidebar a:hover { background: var(--accent); color: #fff; }
.header {
  grid-area: header; background: var(--main);
  display: flex;
  align-items: center; justify-content: space-between;
  padding: 0 1.5rem; box-shadow: 0 2px 4px #0001;
}
.main {
  grid-area: main; background: var(--main);
  padding: 2rem;
}
.main h1 { margin-top: 0; }
.dashboard-widgets {
  display: grid; gap: 1.5rem;
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
  margin-top: 1.5rem;
}
.widget {
  background: var(--bg); border-radius: 10px; padding: 1.2rem;
  box-shadow: 0 2px 8px #0001;
}
.theme-toggle {
  cursor: pointer; padding: 0.5rem 1.2rem;
  border: none; border-radius: 6px;
```

```

    background: var(--accent); color: #fff; font-size: 1rem;
    transition: background .3s;
    margin-left: 2rem;
  }
  @media (max-width: 800px) {
    .dashboard { grid-template-areas: "header" "main" "sidebar"; grid-template-columns:
1fr; grid-template-rows: 60px 1fr auto;}
    .sidebar { flex-direction: row; gap: 2rem; justify-content: center; }
    .main { padding: 1rem;}
  }
</style>
</head>
<body>
<div class="dashboard" id="dashboard">
  <nav class="sidebar">
    <h2>Admin</h2>
    <a href="#" class="active">Dashboard</a>
    <a href="#">Users</a>
    <a href="#">Reports</a>
    <a href="#">Settings</a>
  </nav>
  <header class="header">
    <span>Admin Dashboard</span>
    <button class="theme-toggle" onclick="toggleTheme()">🌙 / 🌞 Theme</button>
  </header>
  <main class="main">
    <h1>Welcome, Admin!</h1>
    <div class="dashboard-widgets">
      <div class="widget">
        <h3>Users</h3>
        <p>Registered: <b>256</b></p>
      </div>
      <div class="widget">
        <h3>Transactions</h3>
        <p>This Month: <b>1,245</b></p>
      </div>
      <div class="widget">
        <h3>Errors</h3>
        <p>Critical: <b>2</b></p>
      </div>
    </div>
  </main>
</div>

```

```
<div class="widget">
  <h3>Revenue</h3>
  <p>$<b>87,500</b></p>
</div>
</div>
</main>
</div>
<script>
function toggleTheme() {
  const theme = document.body.getAttribute('data-theme');
  if(theme === 'dark') {
    document.body.removeAttribute('data-theme');
    localStorage.setItem('theme','light');
  } else {
    document.body.setAttribute('data-theme','dark');
    localStorage.setItem('theme','dark');
  }
}
(function(){ //Persist theme
  if(localStorage.getItem('theme') === 'dark') {
    document.body.setAttribute('data-theme','dark');
  }
})();
</script>
</body>
</html>
```



## OUTPUTS:

### Task-1:

### Student Registration

Name

Email

Age

Register

### Task-2:

#### Basic Interface:

### Your Account

\$0.00

Deposit

Withdraw

Money Deposited:

Your Account

\$1000000.00

Amount

Deposit

Withdraw

Deposit Successful

Money Withdrawal:

Your Account

\$500000.00

Amount

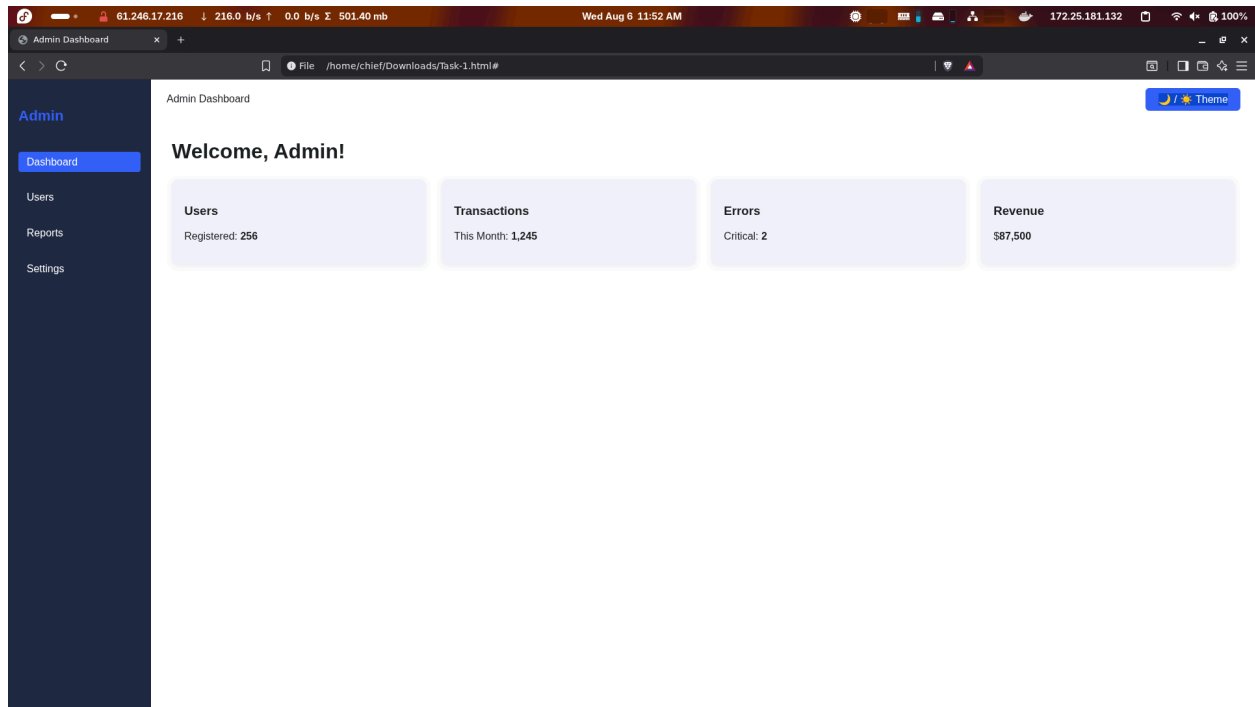
Deposit

Withdraw

Withdrawal Successful

# Task-3:

## Normal Dashboard in Default Mode:



## Normal Dashboard in Dark Mode:

