2 nd code

```
<!DOCTYPE html>
<html>
<head>
  <title>Face Detect</title>
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs"></script>
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow-models/blazeface"></script>
  <style>body{text-align:center}canvas{border:1px solid #f0f0f0}</style>
</head>
<body>



<h3>Upload Image</h3>
<input type="file" id="imgInput"><br><br>
<canvas id="canvas"></canvas>


<script>


let model;
blazeface.load().then(m => model = m);


document.getElementById('imgInput').onchange = async e => {
  if (!model) return alert("Model loading...");


  const file = e.target.files[0];
  const img = new Image();


  img.onload = async () => {
    const c = document.getElementById('canvas');
    const ctx = c.getContext('2d');
```

```
    let scale = Math.min(600 / img.width, 400 / img.height, 1);
    c.width = img.width * scale;
    c.height = img.height * scale;
    ctx.drawImage(img, 0, 0, c.width, c.height);

    const faces = await model.estimateFaces(c, false);
    faces.forEach(f => {
      const [x, y] = f.topLeft, [x2, y2] = f.bottomRight;
      ctx.strokeStyle = 'red';
      ctx.lineWidth = 2;
      ctx.strokeRect(x, y, x2 - x, y2 - y);
    });
  };
  img.src = URL.createObjectURL(file);
};

</script>
</body>
</html>



## Spam
<!DOCTYPE html>
<html>
<head>
<title>Spam Detection</title>
<style>
body{font-family:Arial;padding:20px;text-align:center}
input{width:80%;padding:8px}
button{padding:8px 16px;margin-top:10px}
p{font-weight:bold}
```

```
</style>
</head>
<body>

<h3>Spam Detection</h3>
<input id="msg" placeholder="Enter message"><br>
<button onclick="checkSpam()">Check</button>
<p id="res"></p>

<script>
const spam=['win','free','prize','money','click','lottery','offer','urgent','winner','cash'];

function checkSpam(){
  let m=document.getElementById('msg').value.toLowerCase();
  if(!m){alert('Enter message');return;}
  let found=false;
  for(let w of spam) if(m.includes(w)) found=true;
  let res;
  if(found) res='Spam detected';
  else res='Not Spam';
  document.getElementById('res').innerText=res;
  if(found) document.getElementById('res').style.color='red';
  else document.getElementById('res').style.color='green';
}
</script>

</body>
</html>
```

Sentiment

```
<!DOCTYPE html>
<html>
<head>
<title>Sentiment Analysis</title>
<style>
body{font-family:Arial;padding:20px;text-align:center}
textarea{width:80%;height:100px;padding:8px}
button{padding:8px 16px;margin-top:10px}
p{font-weight:bold}
</style>
</head>
<body>

<h3>Sentiment Analysis</h3>
<textarea id="text" placeholder="Type your feedback..."></textarea><br>
<button onclick="checkSentiment()">Analyze</button>
<p id="result"></p>

<script>
const pos=['good','great','love','happy','awesome','nice'];
const neg=['bad','hate','sad','poor','worst','terrible'];

function checkSentiment(){
  let t=document.getElementById('text').value.toLowerCase();
  if(!t){alert('Enter feedback');return;}
  let p=0,n=0;
```

```
  for(let w of pos) if(t.includes(w)) p++;
  for(let w of neg) if(t.includes(w)) n++;
  let res;
  if(p>n) res='Positive';
  else if(n>p) res='Negative';
  else res='Neutral';
  document.getElementById('result').innerText='Sentiment: '+res;
  if(res==='Positive') document.getElementById('result').style.color='green';
  else if(res==='Negative') document.getElementById('result').style.color='red';
  else document.getElementById('result').style.color='gray';
}
</script>

</body>
</html>
```

```
## 3 and 4
const express = require('express');
const bodyParser = require('body-parser');
const mongoose = require('mongoose');

const app = express();
app.use(bodyParser.json());
const PORT = 3000;

// MongoDB connection
mongoose.connect("mongodb://localhost:27017/testdb")
```

```
  .then(() => console.log('Connected to MongoDB'))
  .catch(err => console.error('Could not connect to MongoDB...', err));

// Schema
const userSchema = new mongoose.Schema({
  uname: { type: String, unique: true, required: true },
  password: { type: String, required: true }
});

const User = mongoose.model('User', userSchema);

// Create user
app.post('/users', async (req, res) => {
  try {
    const user = new User(req.body);
    const savedUser = await user.save();
    res.json(savedUser);
  } catch (err) {
    res.json({ error: err.message });
  }
});

// Get all users
app.get('/users', async (req, res) => {
  try {
    const users = await User.find();
    res.json(users);
  } catch (err) {
    res.status(500).json({ error: err.message });
  }
});
```

```
app.listen(PORT, () => {
   console.log(`Server is Running on port ${PORT}`);
});
```

Text to Speech

```
<!DOCTYPE html>
<html>
<head>
   <title>Text-to-Speech Converter</title>
   <style>
      body { font-family: Arial; padding: 20px; }
      textarea { width: 100%; height: 100px; padding: 10px; margin-top: 10px; }
      button { padding: 10px 20px; margin-top: 10px; cursor: pointer; }
      select { padding: 5px; margin-left: 10px; }
   </style>
</head>
<body>
   <h1>Text-to-Speech Converter</h1>
   <textarea id="text" placeholder="Enter text to speak..."></textarea><br>
   <label for="voice">Voice:</label>
   <select id="voice"></select>
   <button onclick="speakText()">Speak</button>

   <script>
      const synth = window.speechSynthesis;
      const voiceSelect = document.getElementById('voice');

      let voices = [];
```

```
    function populateVoices() {
      voices = synth.getVoices();
      voiceSelect.innerHTML = '';
      voices.forEach((voice, i) => {
        const option = document.createElement('option');
        option.value = i;
        option.text = `${voice.name} (${voice.lang})`;
        voiceSelect.appendChild(option);
      });
    }

    populateVoices();
    if (speechSynthesis.onvoiceschanged !== undefined) {
      speechSynthesis.onvoiceschanged = populateVoices;
    }

    function speakText() {
      const text = document.getElementById('text').value.trim();
      if (!text) { alert("Please enter some text!"); return; }

      const utterance = new SpeechSynthesisUtterance(text);
      const selectedVoice = voices[voiceSelect.value];
      if (selectedVoice) utterance.voice = selectedVoice;

      synth.speak(utterance);
    }
  </script>
</body>
</html>
```

Movie Recommendation

```
<!DOCTYPE html>
<html>
<head>
  <title>Simple Movie Recommendation</title>
</head>
<body>
  <h1>Movie Recommendation System</h1>

  <label>Select your favorite genre:</label>
  <select id="genreSelect">
    <option value="">--Select--</option>
    <option value="Action">Action</option>
    <option value="Comedy">Comedy</option>
    <option value="Horror">Horror</option>
    <option value="Romance">Romance</option>
    <option value="Sci-Fi">Sci-Fi</option>
  </select>

  <label>Select movies you liked before:</label>
  <select id="likedMovies" multiple size="5">
    <option value="Avengers: Endgame">Avengers: Endgame</option>
    <option value="Mad Max: Fury Road">Mad Max: Fury Road</option>
    <option value="The Hangover">The Hangover</option>
    <option value="Superbad">Superbad</option>
    <option value="The Conjuring">The Conjuring</option>
    <option value="It">It</option>
    <option value="The Notebook">The Notebook</option>
    <option value="Pride & Prejudice">Pride & Prejudice</option>
    <option value="Interstellar">Interstellar</option>
```

```
    <option value="Inception">Inception</option>
  </select>

  <button onclick="recommendMovies()">Get Recommendations</button>

  <div id="recommendations"></div>

  <script>
    const movies = [
      { title: "Avengers: Endgame", genre: "Action" },
      { title: "Mad Max: Fury Road", genre: "Action" },
      { title: "The Hangover", genre: "Comedy" },
      { title: "Superbad", genre: "Comedy" },
      { title: "The Conjuring", genre: "Horror" },
      { title: "It", genre: "Horror" },
      { title: "The Notebook", genre: "Romance" },
      { title: "Pride & Prejudice", genre: "Romance" },
      { title: "Interstellar", genre: "Sci-Fi" },
      { title: "Inception", genre: "Sci-Fi" }
    ];

    function recommendMovies() {
      const selectedGenre = document.getElementById("genreSelect").value;
      const likedMovies =
Array.from(document.getElementById("likedMovies").selectedOptions).map(o => o.value);
      const recommendationsDiv = document.getElementById("recommendations");

      if (!selectedGenre) {
        recommendationsDiv.innerHTML = "Please select a genre!";
        return;
      }
```

```
        const recommended = movies
          .filter(m => m.genre === selectedGenre && !likedMovies.includes(m.title));

        if (recommended.length === 0) {
          recommendationsDiv.innerHTML = "No recommendations found.";
        } else {
          recommendationsDiv.innerHTML = "<h3>Recommended Movies:</h3><ul>" +
            recommended.map(m => `<li>${m.title}</li>`).join('') +
            "</ul>";
        }
      }
  </script>
</body>
</html>
```

10th

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <title>E-commerce Recommendations</title>
 <style>
  body { font-family: Arial; background:#f9f9f9; margin:30px; }
  .box { display:inline-block; border:1px solid #ccc; padding:10px; margin:8px;
       width:150px; text-align:center; border-radius:6px; background:#fff; cursor:pointer; }
  .box:hover { background:#eef; }
 </style>
```

```
</head>
<body>

<h2>Products</h2>
<div id="p"></div>

<h2>Recommended for You</h2>
<div id="r">No recommendations yet.</div>

<script>
const products = [
 {name:'Laptop',cat:'Electronics'},{name:'Headphones',cat:'Electronics'},
 {name:'Shoes',cat:'Fashion'},{name:'T-Shirt',cat:'Fashion'},
 {name:'Blender',cat:'Home'},{name:'Vacuum Cleaner',cat:'Home'}
];

let history=[], shown=[];

function render(){
 p.innerHTML='';
 products.forEach(x=>{
  let d=document.createElement('div');
  d.className='box'; d.innerText=`${x.name}\n(${x.cat})`;
  d.onclick=()=>view(x);
  p.appendChild(d);
 });
}

function view(x){
 alert(`You viewed: ${x.name}`);
 history.push(x);
```

```
  recommend();
}

function recommend(){
  let cats=history.map(y=>y.cat);
  let rec=products.filter(y=>cats.includes(y.cat)&&!history.includes(y)&&!shown.includes(y));
  if(!rec.length){ r.innerText='No new recommendations.'; return; }
  r.innerHTML='';
  rec.forEach(z=>{
    let d=document.createElement('div');
    d.className='box'; d.innerText=`${z.name}\n(${z.cat})`;
    r.appendChild(d); shown.push(z);
  });
}

render();
</script>
</body>
</html>
```