

## index.html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

  <title>Quiz Portal - For 9th Mathematics</title>

  <link rel="stylesheet" href="styles.css" />

  <script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.5.1/jspdf.umd.min.js"></script>

</head>

<body>

  <div class="container">

    <h1>Quiz Portal For 9th Mathematics</h1>


    <label for="versionSelect"><strong>Select Version:</strong></label>

    <select id="versionSelect">

      <option value="1">Chapter 1 - Number Systems</option>

      <option value="2">Chapter 2 - Polynomials</option>

      <option value="3">Chapter 3 - Coordinate Geometry</option>

      <option value="4">Chapter 4 - Linear Equations in Two Variables</option>

      <option value="5">Chapter 5 - Introduction to Euclid's Geometry</option>

      <option value="6">Chapter 6 - Lines and Angles</option>

      <option value="7">Chapter 7 - Triangles</option>

      <option value="8">Chapter 8 - Quadrilaterals</option>

      <option value="9">Chapter 9 - Circles</option>

      <option value="10">Chapter 10 - Heron's Formula</option>

    </select>

  </div>

</body>

</html>
```

```
<option value="11">Chapter 11 - Surface Areas and Volumes</option>
<option value="12">Chapter 12 - Statistics</option>
<option value="13">Chapters 1 to 6</option>
<option value="14">Chapters 7 to 12</option>
</select>
<button onclick="loadQuiz()">Load Quiz</button>

<div id="quiz"></div>

<button onclick="submitQuiz()">Submit Answers</button>
<button id="downloadBtn" disabled>Download PDF</button>
<button onclick="retakeQuiz()">Retake Quiz</button>

<div id="result"></div>
<div id="explanation"></div>
</div>

<script src="script.js"></script>
</body>
</html>
```

### **script.js**

```
let currentVersion = "1";
let storageKey = "";
let shuffledQuiz = [];
```

```
const versionTitles = {  
  "1": "Chapter 1 - Number System",  
  "2": "Chapter 2 - Polynomials",  
  "3": "Chapter 3 - Coordinate Geometry",  
  "4": "Chapter 4 - Linear Equations",  
  "5": "Chapter 5 - Euclid's Geometry",  
  "6": "Chapter 6 - Lines and Angles",  
  "7": "Chapter 7 - Triangles",  
  "8": "Chapter 8 - Quadrilaterals",  
  "9": "Chapter 9 - Circles",  
  "10": "Chapter 10 - Heron's Formula",  
  "11": "Chapter 11 - Surface Areas and Volumes",  
  "12": "Chapter 12 - Statistics",  
  "13": "Chapters 1 to 6",  
  "14": "Chapters 7 to 12"  
};
```

```
let logoBase64 = null;
```

```
let stampBase64 = null;
```

```
// Load logo
```

```
const logoImage = new Image();
```

```
logoImage.src = "logo.jpeg";
```

```
logoImage.onload = () => {
```

```
  const canvas = document.createElement("canvas");
```

```
  canvas.width = logoImage.width;
```

```
canvas.height = logoImage.height;

const ctx = canvas.getContext("2d");

ctx.drawImage(logoImage, 0, 0);

logoBase64 = canvas.toDataURL("image/jpeg");

};
```

```
// Load stamp

const stampImage = new Image();

stampImage.src = "stamp.jpeg";

stampImage.onload = () => {

  const canvas = document.createElement("canvas");

  canvas.width = stampImage.width;

  canvas.height = stampImage.height;

  const ctx = canvas.getContext("2d");

  ctx.drawImage(stampImage, 0, 0);

  stampBase64 = canvas.toDataURL("image/jpeg");

};
```

```
function shuffleArray(array) {

  for (let i = array.length - 1; i > 0; i--) {

    const j = Math.floor(Math.random() * (i + 1));

    [array[i], array[j]] = [array[j], array[i]];

  }

}
```

```
async function loadQuiz() {
```

```
const version = document.getElementById("versionSelect").value;
currentVersion = version;
storageKey = `quizResults_v${version}`;
```

```
document.getElementById("result").textContent = "";
document.getElementById("explanation").innerHTML = "";
document.getElementById("downloadBtn").disabled = true;
```

```
try {
  const response = await fetch(`quizzes/quiz_v${version}.json`);
  const quiz = await response.json();
  shuffledQuiz = JSON.parse(JSON.stringify(quiz));
```

```
  shuffledQuiz.forEach(q => {
    const correctVal = q.options[q.correct];
    shuffleArray(q.options);
    q.correct = q.options.indexOf(correctVal);
  });
  shuffleArray(shuffledQuiz);
```

```
const quizContainer = document.getElementById("quiz");
quizContainer.innerHTML = "";
shuffledQuiz.forEach((q, i) => {
  let html = `

<strong>${i + 1}. ${q.question}</strong></p>`;
  q.options.forEach((opt, j) => {


```

```
html += `<label><input type="radio" name="question${i}" value="${j}">
${opt}</label><br>`;
```

```
});
```

```
quizContainer.innerHTML += html + "<br>";
```

```
});
```

```
localStorage.removeItem(storageKey);
```

```
} catch (error) {
```

```
document.getElementById("quiz").innerHTML = "Error loading quiz. Please check the file.";
```

```
console.error("Fetch error:", error);
```

```
}
```

```
}
```

```
function submitQuiz() {
```

```
let score = 0;
```

```
const responses = [];
```

```
let explanationHTML = "<h2>Explanations:</h2>";
```

```
shuffledQuiz.forEach((q, i) => {
```

```
const selected = document.querySelector(`input[name="question${i}"]:checked`);
```

```
const selectedIndex = selected ? parseInt(selected.value) : null;
```

```
const isCorrect = selectedIndex === q.correct;
```

```
responses.push({
```

```
question: q.question,
```

```
selected: selectedIndex !== null ? q.options[selectedIndex] : "Not answered",
```

```

    correct: q.options[q.correct],
    explanation: q.explanation,
    isCorrect
  });

  if (isCorrect) score++;
});

document.getElementById("result").textContent = `You scored ${score} out of
${shuffledQuiz.length}!`;

responses.forEach((res, i) => {
  explanationHTML += `<p><strong>${i + 1}. ${res.question}</strong><br>
  Your answer: ${res.selected}<br>
  Correct answer: ${res.correct}<br>
  Explanation: ${res.explanation}</p>`;
});

document.getElementById("explanation").innerHTML = explanationHTML;
localStorage.setItem(storageKey, JSON.stringify({ score, responses }));
document.getElementById("downloadBtn").disabled = false;
}

function generatePDF() {
  const data = JSON.parse(localStorage.getItem(storageKey));
  if (!data) {

```

```
    alert("Please submit the quiz first.");  
    return;  
}
```

```
const { jsPDF } = window.jspdf;  
const doc = new jsPDF({ unit: "mm", format: "a4", orientation: "portrait" });
```

```
const title = versionTitles[currentVersion] || `Quiz v${currentVersion}`;  
const safeTitle = title.replace(/[\V:*?"<>|]/g, "-");  
const now = new Date();  
const dateString = now.toLocaleDateString("en-GB", {  
    day: "2-digit", month: "short", year: "numeric"  
}).replace(/ /g, "-");
```

```
const pageHeight = doc.internal.pageSize.getHeight();  
const pageWidth = doc.internal.pageSize.getWidth();  
const marginLeft = 15;  
const marginRight = 15;  
const usableWidth = pageWidth - marginLeft - marginRight;  
let y = 20;
```

```
// Top-left logo  
if (logoBase64) {  
    doc.drawImage(logoBase64, "JPEG", marginLeft, 10, 30, 20);  
}
```



```
// Title and score

doc.setFont("helvetica", "bold");

doc.setFontSize(14);

doc.text(`${title} - Quiz Results`, marginLeft + 35, 15);

doc.setFontSize(11);

doc.text(`Date: ${now.toDateString()}`, marginLeft + 35, 22);

doc.text(`Score: ${data.score} / ${data.responses.length}`, marginLeft + 35, 28);

y = 35;


doc.setFont("helvetica", "normal");

doc.setFontSize(11);


data.responses.forEach((res, index) => {

  const block = [

    `${index + 1}. ${res.question}`,

    `Your answer: ${res.selected}`,

    `Correct answer: ${res.correct}`,

    `Explanation: ${res.explanation}`

  ];

  block.forEach(line => {

    const wrapped = doc.splitTextToSize(line, usableWidth);

    if (y + wrapped.length * 6 > pageHeight - 50) {

      doc.addPage();

      y = 20;

    }

  });

});
```

```
    doc.text(wrapped, marginLeft, y);
    y += wrapped.length * 6;
  });
  y += 4;
});

// Stamp + Signature
if (y + 40 > pageHeight - 20) {
  doc.addPage();
  y = 20;
}

if (stampBase64) {
  const stampWidth = 30;
  const stampX = (pageWidth - stampWidth) / 2;
  doc.addImage(stampBase64, "JPEG", stampX, y, stampWidth, 30);
  y += 35;
}

doc.setFont("helvetica", "italic");
doc.text("_____", (pageWidth - 60) / 2, y);
doc.text("Teacher's Signature", (pageWidth - 60) / 2, y + 6);

doc.save(`${safeTitle} (${dateString}).pdf`);
}
```

```
function retakeQuiz() {
  loadQuiz();
}

document.addEventListener("DOMContentLoaded", () => {
  const btn = document.getElementById("downloadBtn");
  if (btn) btn.addEventListener("click", generatePDF);
});
```

-X-

Analyse this code and add some more features in the code. 1. in generatePDF make A4 Printout styled Top-Bottom and Left-Right Margins for taking printouts of quizResult. 2. Update the code for wrapping the contents of quizResult within A4 styled margins for printouts. 3. Make one more feature adjustment in code for entering the student name as input before attempting the quiz using Load Quiz so that students name also appear below the quiz title inside the downloadable pdf

Great work so far, Jagdev! Your existing quiz application is functionally sound and you're already handling user responses, explanations, and PDF generation effectively. Now let's enhance it with the three requested features:

#### ✅ 1 & 2: Add Proper A4 Printout Margins and Wrap Content Accordingly

In the generatePDF() function, you already calculate margins and wrap text using doc.splitTextToSize(). You'll just need to:

- Standardize **top (20mm), bottom (20mm), left/right (15mm)** margins.
- Use doc.splitTextToSize() on *all* text blocks so content respects these margins.
- Update line spacing logic to avoid overflow beyond printable area.

Here's how you can refine your layout logic:

javascript

```
const marginTop = 20;
const marginBottom = 20;
const marginLeft = 15;
const marginRight = 15;
const usableWidth = pageWidth - marginLeft - marginRight;
let y = marginTop;
```

When looping through content (e.g. response blocks):

javascript

```
const wrapped = doc.splitTextToSize(line, usableWidth);
if (y + wrapped.length * 6 > pageHeight - marginBottom) {
  doc.addPage();
  y = marginTop;
}
doc.text(wrapped, marginLeft, y);
y += wrapped.length * 6 + 4;
```

### ✅ 3: Add Student Name Input & Include in PDF

To collect the student's name and include it in the PDF:

#### a. Add Input Field in index.html

html

```
<label for="studentName"><strong>Student Name:</strong></label>
<input type="text" id="studentName" placeholder="Enter your name" /><br><br>
```

Place this just above the version selector.

#### b. Store Student Name in JavaScript

Update submitQuiz():

javascript

```
const studentName = document.getElementById("studentName").value.trim();  
localStorage.setItem("studentName", studentName);
```

### **c. Use Student Name in PDF**

Update generatePDF():

javascript

```
const studentName = localStorage.getItem("studentName") || "Unnamed Student";  
doc.text(`Name: ${studentName}`, marginLeft + 35, 34);  
y = 40; // Adjust Y position accordingly
```

Also, update where you set y before starting the responses block.

Once implemented, this will give your quiz PDF a polished appearance for printing and add personalized headers for each student.

Would you like help integrating a preview feature so students can see their results before downloading the PDF? That could be a neat touch for your portal.

Can you explain the code changes more in-depth? What other features could I add to my quiz app? Can you help with specific code adjustments?