index.html

```
<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width,initial-scale=1">

<title>EduBoard — Role Selection</title>

<style>

:root{--bg:#f4f7fb;--card:#fff;--accent:#0ea5a4;--muted:#6b7280;--
glass:rgba(255,255,255,0.7)}

*{box-sizing:border-box;font-family:Inter,ui-sans-serif,system-ui,Segoe
UI,Roboto,"Helvetica Neue",Arial}

body{margin:0;background:linear-gradient(180deg,#f8fafc, #eef2f7);min-
height:100vh;display:flex;align-items:center;justify-
content:center;padding:20px}

.container{width:960px;background:var(--card);border-radius:14px;box-
shadow:0 10px 30px rgba(16,24,40,0.08);overflow:hidden;display:grid;grid-
template-columns:1fr 420px}

.brand{padding:40px;background:linear-gradient(135deg,#0ea5a4,
#3b82f6);color:white;display:flex;flex-direction:column;gap:18px}

.brand h1{margin:0;font-size:28px;letter-spacing:0.4px}

.brand p{margin:0;opacity:0.95}

.actions{padding:40px}

.role-card{display:flex;flex-direction:column;gap:14px}

.role{background:var(--bg);padding:18px;border-
radius:10px;cursor:pointer;border:1px solid transparent;transition:all
.18s;display:flex;justify-content:space-between;align-items:center}

.role:hover{transform:translateY(-4px);box-shadow:0 8px 20px
rgba(2,6,23,0.06)}
```

```
.role strong{font-size:16px}

.role small{color:var(--muted)}

.btn{display:inline-block;padding:10px 16px;border-radius:8px;background:var(--accent);color:white;text-decoration:none;margin-top:18px}

.footer{font-size:13px;color:var(--muted);margin-top:18px}

.logo{background:rgba(255,255,255,0.15);padding:8px;border-radius:8px;width:max-content}

</style>

</head>

<body>

<div class="container" role="main">

  <div class="brand">

    <div class="logo"><strong>EduBoard</strong></div>

    <h1>Welcome to EduBoard</h1>

    <p>📚 Why Students Love EduBoard

Imagine a classroom where learning feels like play! On EduBoard, every quiz you take brings you closer to unlocking exciting games like Sudoku, Word Match, and Memory Pairs. No boring lessons here — just challenges, rewards, and the thrill of beating your own best score. It’s school, but upgraded into a fun digital adventure!</p>

    <p style="font-size:17px;opacity:0.9">👩‍🏫 Why Teachers Trust EduBoard

EduBoard is more than a teaching tool — it’s your smart classroom partner. Upload notes, design quizzes in minutes, and watch as students stay motivated with game-based rewards. While you focus on teaching, EduBoard handles the engagement. It’s education that’s not only effective but also unforgettable.</p>

  </div>


  <div class="actions">
```

```
  <h2>Select your role</h2>
  <div class="role-card">
    <div class="role" onclick="go('student_register.html')">
      <div>
        <strong>Register as Student</strong><br><small>Create student account with class & roll number</small>
      </div>
      <div><span>&rsaquo;</span></div>
    </div>
    <div class="role" onclick="go('teacher_register.html')">
      <div>
        <strong>Register as Teacher</strong><br><small>Create teacher account to upload notes & create quizzes</small>
      </div>
      <div><span>&rsaquo;</span></div>
    </div>

    <a class="btn" href="login.html">Already registered? Login</a>

   <div class="footer">
  <p>✨ EduBoard — Where knowledge meets fun, and every student’s journey becomes extraordinary ✨</p>
  <p>© 2025 EduBoard. All Rights Reserved.</p>
</div>

<style>
.footer {
```

```
  background: linear-gradient(90deg, #2563eb, #1e3a8a);
  color: white;
  text-align: center;
  padding: 20px 10px;
  font-family: Inter, Segoe UI, Arial, sans-serif;
  border-top-left-radius: 20px;
  border-top-right-radius: 20px;
  box-shadow: 0 -4px 12px rgba(0,0,0,0.2);
}
.footer p {
  margin: 6px 0;
}
.footer p:first-child {
  font-size: 16px;
  letter-spacing: 0.5px;
}
.footer p:last-child {
  font-size: 13px;
  opacity: 0.8;
}
</style>

    </div>
  </div>
</div>
```

```
<script>
function go(h){ window.location.href = h; }
</script>
</body>
</html>
```

login.html

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,initial-scale=1">
<title>EduBoard — Login</title>
<style>
:root{--bg:#fff;--accent:#ef4444;--muted:#556577}
body{margin:0;font-family:Inter,system-ui,Segoe UI,Roboto,Arial;background:linear-gradient(180deg,#fbfdff,#ffffff);display:flex;align-items:center;justify-content:center;padding:24px}
.card{width:560px;background:white;border-radius:12px;padding:26px;box-shadow:0 12px 30px rgba(16,24,40,0.06)}
h1{margin:0 0 12px 0}
.input{display:flex;flex-direction:column;margin-bottom:12px}
label{font-size:13px;color:var(--muted);margin-bottom:6px}
input{padding:10px;border-radius:8px;border:1px solid #eee;outline:none}
.row{display:flex;gap:10px;align-items:center}
```

```
.btn{padding:10px 14px;border-radius:8px;border:none;background:#10b981;color:white;cursor:pointer}
.small{font-size:13px;color:var(--muted)}
</style>
</head>
<body>
<div class="card">
  <h1>Login</h1>
  <div class="small">Login with your registered email & password</div>

  <div style="margin-top:16px">
    <div class="input"><label>Email</label><input id="email" type="email"></div>
    <div class="input"><label>Password</label><input id="password" type="password"></div>
    <div style="display:flex;justify-content:space-between;align-items:center">
      <button class="btn" onclick="login()">Login</button>
      <div><a href="index.html">Back to role selection</a></div>
    </div>
  </div>
</div>

<script>
function getUsers(){return JSON.parse(localStorage.getItem('edu_users')||'[]');}
function login(){
  const email=document.getElementById('email').value.trim();
  const pwd=document.getElementById('password').value;
```

```
  const users=getUsers();
  const u=users.find(x=>x.email===email && x.password===pwd);
  if(!u){ alert('Invalid credentials'); return; }
  // if teacher and no assignedClass yet, ask and save
  if(u.role==='teacher' && !u.assignedClass){
    const cls = prompt('Enter the class (number) you teach (example: 9) — this binds your uploads/quizzes to that class.');
    if(cls){ u.assignedClass = cls.toString(); const others = users.map(x=> x.email===u.email? u : x); localStorage.setItem('edu_users', JSON.stringify(others)); }
  }
  // set session
  localStorage.setItem('edu_session', JSON.stringify({email:u.email,role:u.role,name:u.name,cls:u.cls||null}));
  alert('Login successful! Redirecting to dashboard...');
  if(u.role==='student') location.href='student_dashboard.html'; else location.href='teacher_dashboard.html';
}
</script>
</body>
</html>
```

student_register.html

```
<!doctype html>
<html lang="en">
<head>
```

```
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,initial-scale=1">
<title>EduBoard — Student Registration</title>
<style>
:root{--bg:#f7fbff;--card:#fff;--accent:#2563eb;--muted:#556577}
body{margin:0;font-family:Inter,system-ui,Segoe UI,Roboto,Arial;background:linear-gradient(180deg,#f0f6ff,#ffffff);display:flex;align-items:center;justify-content:center;padding:24px}
.card{width:760px;background:var(--card);border-radius:12px;padding:26px;box-shadow:0 12px 30px rgba(16,24,40,0.06)}
h1{margin:0 0 8px 0}
.form{display:grid;grid-template-columns:1fr 1fr;gap:12px}
.input{display:flex;flex-direction:column}
label{font-size:13px;color:var(--muted);margin-bottom:6px}
input,select{padding:10px;border-radius:8px;border:1px solid #e6eef9;outline:none}
.actions{grid-column:1 / -1;display:flex;gap:10px;align-items:center;justify-content:flex-end}
.btn{padding:10px 14px;border-radius:8px;border:none;background:var(--accent);color:white;cursor:pointer}
.note{font-size:13px;color:var(--muted)}
.small{font-size:12px;color:#f05454}
</style>
</head>
<body>
<div class="card">
  <h1>Student Registration</h1>
```

```
 <p class="note">Fill details.</p>

 <form id="stuForm" class="form" onsubmit="return registerStudent(event)">
  <div class="input">
   <label>Full Name</label>
   <input id="name" required>
  </div>
  <div class="input">
   <label>Roll Number</label>
   <input id="roll" required>
  </div>
  <div class="input">
   <label>Class (currently studying)</label>
   <select id="classSelect" required>
    <option value="">Select class</option>
    <option>6</option><option>7</option><option>8</option><option>9</o
ption><option>10</option><option>11</option><option>12</option>
   </select>
  </div>
  <div class="input">
   <label>Email ID</label>
   <input id="email" type="email" required>
  </div>
  <div class="input">
   <label>Mobile Number</label>
   <input id="mobile" required pattern="\d*">
  </div>
```

```
    <div class="input">
      <label>School Name</label>
      <input id="school" required>
    </div>
    <div class="input">
      <label>Password</label>
      <input id="password" type="password" required minlength="4">
    </div>
    <div class="input">
      <label>Confirm Password</label>
      <input id="password2" type="password" required>
    </div>

    <div class="actions">
      <a href="index.html" style="text-decoration:none;padding:8px 12px;border-radius:8px;border:1px solid #e6eef9;color:var(--muted)">Back</a>
      <button class="btn" type="submit">Register Student</button>
    </div>
  </form>
</div>

<script>
function getUsers(){return JSON.parse(localStorage.getItem('edu_users')||'[]');}
function saveUsers(u){localStorage.setItem('edu_users', JSON.stringify(u));}

function registerStudent(e){
  e.preventDefault();
```

```
  const name=document.getElementById('name').value.trim();
  const roll=document.getElementById('roll').value.trim();
  const cls=document.getElementById('classSelect').value;
  const email=document.getElementById('email').value.trim();
  const mobile=document.getElementById('mobile').value.trim();
  const school=document.getElementById('school').value.trim();
  const pwd=document.getElementById('password').value;
  const pwd2=document.getElementById('password2').value;
  if(pwd!==pwd2){alert('Passwords do not match'); return;}
  let users=getUsers();
  if(users.find(x=>x.email===email)){alert('Email already registered');return;}
  const
user={role:'student',name,roll,cls,email,mobile,school,password:pwd,registere
dAt:new Date().toISOString()};
  users.push(user);
  saveUsers(users);
  // popup message
  alert('Registration successful! .');
  // prepare mailto and sms links (demo)
  const subject=encodeURIComponent('EduBoard — Registration
Confirmation');
  const body=encodeURIComponent(`Hello ${name},\n\nYour EduBoard
student account is created. Login using ${email}.\n\nRegards, EduBoard`);
  const mailto=`mailto:${email}?subject=${subject}&body=${body}`;
  const sms=`sms:${mobile}?body=${encodeURIComponent('Your EduBoard
account is created. Login using your email.')}`;
  // open mail client in new tab (demo)
  window.open(mailto);
```

```
  // open sms (mobile browsers may handle)
  //window.open(sms);
  // redirect to login
  setTimeout(()=> location.href='login.html', 400);
  return false;
}
</script>
</body>
</html>
```

teacher_register.html

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,initial-scale=1">
<title>EduBoard — Teacher Registration</title>
<style>
:root{--bg:#fffaf0;--card:#fff;--accent:#8b5cf6;--muted:#6b7280}
body{margin:0;font-family:Inter,system-ui,Segoe UI,Roboto,Arial;background:linear-gradient(180deg,#fff,#f7f7ff);display:flex;align-items:center;justify-content:center;padding:24px}
.card{width:720px;background:var(--card);border-radius:12px;padding:26px;box-shadow:0 12px 30px rgba(16,24,40,0.06)}
h1{margin:0 0 8px 0}
```

```
.form{display:grid;grid-template-columns:1fr 1fr;gap:12px}
.input{display:flex;flex-direction:column}
label{font-size:13px;color:var(--muted);margin-bottom:6px}
input,select{padding:10px;border-radius:8px;border:1px solid #eee;outline:none}
.actions{grid-column:1 / -1;display:flex;gap:10px;align-items:center;justify-content:flex-end}
.btn{padding:10px 14px;border-radius:8px;border:none;background:var(--accent);color:white;cursor:pointer}
.note{font-size:13px;color:var(--muted)}
</style>
</head>
<body>
<div class="card">
  <h1>Teacher Registration</h1>
  <p class="note">Create teacher account. You will be asked at first login for which class you teach — that ties your uploads and quizzes to that class.</p>

  <form id="teaForm" class="form" onsubmit="return registerTeacher(event)">
    <div class="input">
      <label>Full Name</label>
      <input id="tname" required>
    </div>
    <div class="input">
      <label>Email ID</label>
      <input id="temail" type="email" required>
    </div>
```

```
    <div class="input">
      <label>Mobile Number</label>
      <input id="tmobile" required pattern="\d*">
    </div>
    <div class="input">
      <label>Highest Qualification</label>
      <input id="qual" required>
    </div>
    <div class="input">
      <label>School Name</label>
      <input id="tschool" required>
    </div>
    <div class="input">
      <label>Password</label>
      <input id="tpassword" type="password" required minlength="4">
    </div>

    <div class="actions">
      <a href="index.html" style="text-decoration:none;padding:8px 12px;border-
radius:8px;border:1px solid #eee;color:var(--muted)">Back</a>
      <button class="btn" type="submit">Register Teacher</button>
    </div>
  </form>
</div>

<script>
function getUsers(){return JSON.parse(localStorage.getItem('edu_users')||'[]');}
```

```
function saveUsers(u){localStorage.setItem('edu_users', JSON.stringify(u));}

function registerTeacher(e){
  e.preventDefault();
  const name=document.getElementById('tname').value.trim();
  const email=document.getElementById('temail').value.trim();
  const mobile=document.getElementById('tmobile').value.trim();
  const qual=document.getElementById('qual').value.trim();
  const school=document.getElementById('tschool').value.trim();
  const pwd=document.getElementById('tpassword').value;
  let users=getUsers();
  if(users.find(x=>x.email===email)){alert('Email already registered');return;}
  const
user={role:'teacher',name,email,mobile,qual,school,password:pwd,registeredA
t:new Date().toISOString(),assignedClass:null};
  users.push(user);
  saveUsers(users);
  alert('Teacher registered! ');
  const subject=encodeURIComponent('EduBoard — Teacher Registration');
  const body=encodeURIComponent(`Hello ${name},\nYour teacher account is
created. Login using ${email}.\nRegards, EduBoard`);
  window.open(`mailto:${email}?subject=${subject}&body=${body}`);
  setTimeout(()=> location.href='login.html', 400);
  return false;
}
</script>
</body>
```

</html>

student_dashboard.html

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8"><meta name="viewport" content="width=device-width,initial-scale=1">

<title>EduBoard — Student Dashboard</title>

<style>

:root{--card:#fff;--accent:#0ea5a4;--muted:#6b7280}

body{margin:0;font-family:Inter,system-ui,Segoe UI,Roboto,Arial;background:linear-gradient(180deg,#f7fbff,#ffffff);min-height:100vh}

.header{background:linear-gradient(90deg,#0ea5a4,#3b82f6);color:white;padding:18px 26px;display:flex;justify-content:space-between;align-items:center}

.container{padding:20px;max-width:1100px;margin:0 auto}

.grid{display:grid;grid-template-columns:2fr 1fr;gap:18px}

.card{background:var(--card);padding:16px;border-radius:12px;box-shadow:0 8px 20px rgba(16,24,40,0.04)}

.h2{margin:0 0 8px 0}

.quiz{border-radius:8px;padding:12px;border:1px dashed #e6eef9;margin-bottom:10px;display:flex;justify-content:space-between;align-items:center}

.btn{padding:8px 12px;border-radius:8px;border:none;background:#2563eb;color:white;cursor:pointer}

```
.small{font-size:13px;color:var(--muted)}
.note{font-size:13px;color:var(--muted);margin-bottom:10px}
</style>
</head>
<body>
<div class="header">
  <div><strong>EduBoard</strong> — Student Dashboard</div>
  <div id="userbox"></div>
</div>

<div class="container">
  <div class="grid">
    <div>
      <div class="card">
        <h3 class="h2">Available Quizzes</h3>
        <div id="quizzesArea" class="note">Loading quizzes...</div>
      </div>

      <div class="card" style="margin-top:14px">
        <h3 class="h2">Your Results</h3>
        <div id="resultsArea" class="note">No results yet.</div>
      </div>
    </div>

    <div>
      <div class="card">
```

```
      <h3 class="h2">Notes & Resources</h3>
      <div id="notesArea" class="note">No notes uploaded.</div>
    </div>

    <div class="card" style="margin-top:14px">
      <h3 class="h2">Unlocked Games</h3>
      <div id="gamesArea" class="note">Score quizzes to unlock games. <br><a href="games.html">Open Games Page</a></div>
    </div>
  </div>
 </div>
</div>

<!-- Quiz modal -->
<div id="quizModal" style="display:none;position:fixed;inset:0;background:rgba(2,6,23,0.5);align-items:center;justify-content:center">
 <div style="background:white;padding:18px;border-radius:10px;width:760px;max-height:90vh;overflow:auto">
  <h3 id="qtitle">Quiz</h3>
  <div id="qmeta" class="small"></div>
  <form id="quizForm"></form>
  <div style="display:flex;justify-content:space-between;align-items:center;margin-top:12px">
    <div id="timer" class="small">10:00</div>
    <div>
      <button class="btn" onclick="submitQuiz()">Submit</button>
```

```
      <button class="btn" style="background:#ef4444"
onclick="closeModal()">Close</button>
    </div>
   </div>
  </div>
</div>

<script>
function getSession(){return
JSON.parse(localStorage.getItem('edu_session')||'null');}
function getUsers(){return JSON.parse(localStorage.getItem('edu_users')||'[]');}
function getQuizzes(){return
JSON.parse(localStorage.getItem('edu_quizzes')||'[]');}
function getNotes(){return
JSON.parse(localStorage.getItem('edu_notes')||'[]');}
function getResults(){return
JSON.parse(localStorage.getItem('edu_results')||'[]');}
function saveResults(r){localStorage.setItem('edu_results', JSON.stringify(r));}

const sess = getSession();
if(!sess){ alert('No active session. Please login.'); location.href='login.html'; }
document.getElementById('userbox').innerHTML =
`<div><strong>${sess.name}</strong><div class="small">Class: ${sess.cls ||
'N/A'}</div></div>`;

function renderQuizzes(){
  const all = getQuizzes();
  const area = document.getElementById('quizzesArea');
```

```
  const now = new Date();
  const myClass = sess.cls;
  let out='';
  const visible = all.filter(q=> q.targetClass===myClass);
  if(!visible.length){ area.innerHTML = 'Nothing uploaded for your class yet.'; return; }
  visible.forEach(q=>{
    const posted = new Date(q.postedAt);
    const diff = (now - posted);
    const hrs = diff/1000/60/60;
    const available = hrs<=24;
    out += `<div class="quiz"><div>
      <strong>${q.subject}</strong><div class="small">By: ${q.teacherName} — ${q.teacherMobile || ''} | Posted: ${posted.toLocaleString()}</div>
      <div class="small">Status: ${available? 'Available' : 'Closed for attempts'}</div>
    </div>
    <div>`;
    if(available){
      out += `<button class="btn" onclick="openQuiz('${q.id}')">Attempt</button>`;
    } else {
      out += `<button class="btn" disabled>Quiz Closed</button>`;
    }
    out += `</div></div>`;
  });
  area.innerHTML = out;
```

```
}

function renderNotes(){
  const notes = getNotes();
  const area = document.getElementById('notesArea');
  const myClass = sess.cls;
  const visible = notes.filter(n=> n.targetClass===myClass);
  if(!visible.length){ area.innerHTML = 'Nothing uploaded for your class yet.';
return; }
  let out='';
  visible.forEach(n=>{
    const posted = new Date(n.postedAt);
    out += `<div style="margin-bottom:8px"><strong>${n.title}</strong><div
class="small">${n.teacherName} — ${posted.toLocaleString()}</div>`;
    if(n.pdfDataUrl){
      out += `<div><a href="${n.pdfDataUrl}"
download="${n.title}.pdf">Download PDF</a></div>`;
    }
    if(n.text) out += `<div class="small">${n.text.substring(0,220)}</div>`;
    out += `</div>`;
  });
  area.innerHTML = out;
}

function renderResults(){
  const res = getResults().filter(r=> r.studentEmail === sess.email);
  const area = document.getElementById('resultsArea');
```

```
  if(!res.length){ area.innerHTML = 'No quiz attempts yet.'; return; }
  res.sort((a,b)=> b.score - a.score || a.timeTaken - b.timeTaken);
  let out='<ol>';
  res.forEach(r=>{
    out += `<li><strong>${r.subject}</strong> — Score: ${r.score}/10 — Time: ${r.timeTaken}s — ${new Date(r.at).toLocaleString()}</li>`;
  });
  out += '</ol>';
  area.innerHTML = out;

  // keep for history, but unlock will be based on today's quiz
  const best = Math.max(...res.map(x=>x.score),0);
  localStorage.setItem('edu_games_unlocked', best);
  document.getElementById('gamesArea').innerHTML = `Best score: ${best}. <br><a href="games.html">Open Games Page</a>`;
}

renderQuizzes();
renderNotes();
renderResults();

function openQuiz(id){
  const q = getQuizzes().find(x=>x.id===id);
  if(!q) { alert('Quiz not found'); return; }
  document.getElementById('qtitle').innerText = q.subject + ' — Quiz';
```

```
  document.getElementById('qmeta').innerText = `Posted by ${q.teacherName} (${q.teacherMobile || 'no mobile'}). Posted at ${new Date(q.postedAt).toLocaleString()}`;
  const form = document.getElementById('quizForm');
  form.innerHTML = '';
  q.questions.forEach((qu,idx)=>{
    const html = `<div style="margin-bottom:8px"><strong>Q${idx+1}.</strong> ${qu.q}<div style="margin-top:6px">
      ${qu.options.map((op,i)=>`<div><label><input type="radio" name="q${idx}" value="${i}"> ${op}</label></div>`).join('')}
    </div></div>`;
    form.insertAdjacentHTML('beforeend', html);
  });
  showModal();
  startTimer(10*60, () => { alert('Time up! Submitting...'); submitQuiz(id); });
  form.dataset.quizId = id;
}

let timerInterval=null;
function startTimer(seconds, onExpire){
  clearInterval(timerInterval);
  let s=seconds;
  updateTimerDisplay(s);
  timerInterval=setInterval(()=>{
    s--; if(s<0){ clearInterval(timerInterval); onExpire(); return; }
    updateTimerDisplay(s);
  },1000);
```

```
  document.getElementById('timer').dataset.rem = s;
}
function updateTimerDisplay(s){
  const m = Math.floor(s/60); const sec = s%60;
  document.getElementById('timer').innerText =
`${String(m).padStart(2,'0')}:${String(sec).padStart(2,'0')}`;
  document.getElementById('timer').dataset.rem = s;
}

function showModal(){
document.getElementById('quizModal').style.display='flex'; }
function closeModal(){
document.getElementById('quizModal').style.display='none';
clearInterval(timerInterval); }

function submitQuiz(forcedId){
  const form=document.getElementById('quizForm');
  const qid = forcedId || form.dataset.quizId;
  const quiz = getQuizzes().find(x=>x.id===qid);
  if(!quiz) { alert('Quiz not loaded'); return; }
  let score=0;
  const startRem = parseInt(document.getElementById('timer').dataset.rem ||
'0');
  const timeTaken = 10*60 - startRem;
  quiz.questions.forEach((qu,idx)=>{
    const sel = form.querySelector(`input[name="q${idx}"]:checked`);
    const val = sel ? parseInt(sel.value) : null;
    if(val!==null && val===qu.correct) score++;
```

```
  });
  const results = getResults();
  results.push({quizId:qid,studentEmail:sess.email,studentName:sess.name,subject:quiz.subject,score,at:new Date().toISOString(),timeTaken});
  saveResults(results);

  // ☑ Save today's quiz record for daily game unlock
  const today=new Date().toISOString().split("T")[0];
  localStorage.setItem("edu_quiz_record",JSON.stringify({score:score,date:today}));

  alert('Submitted. Your score: ' + score + '/10');
  closeModal();
  renderResults();
  renderQuizzes();
}
</script>
</body>
</html>
```

teacher_dashboard.html

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8"><meta name="viewport" content="width=device-width,initial-scale=1">
```

<title>EduBoard — Teacher Dashboard</title>

<style>

:root{--card:#fff;--accent:#8b5cf6;--muted:#6b7280}

body{margin:0;font-family:Inter,system-ui,Segoe UI,Roboto,Arial;background:linear-gradient(180deg,#fff7ff,#ffffff);min-height:100vh}

.header{background:linear-gradient(90deg,#8b5cf6,#06b6d4);color:white;padding:18px 26px;display:flex;justify-content:space-between;align-items:center}

.container{padding:20px;max-width:1100px;margin:0 auto}

.grid{display:grid;grid-template-columns:1fr 1fr;gap:18px}

.card{background:var(--card);padding:16px;border-radius:12px;box-shadow:0 8px 20px rgba(16,24,40,0.04)}

.h2{margin:0 0 8px 0}

.input{display:flex;flex-direction:column;margin-bottom:10px}

label{font-size:13px;color:var(--muted);margin-bottom:6px}

input,select,textarea{padding:8px;border-radius:8px;border:1px solid #eee;outline:none}

.btn{padding:8px 12px;border-radius:8px;border:none;background:#10b981;color:white;cursor:pointer}

.small{font-size:13px;color:var(--muted)}

</style>

</head>

<body>

EduBoard — Teacher Dashboard

```
<div class="container">
  <div class="grid">
    <div>
      <div class="card">
        <h3 class="h2">Create Quiz (Max 10 questions)</h3>
        <div class="small">When students attempt, quiz is available for 24 hours from posted time. You must add exactly 10 questions (or up to 10).</div>
        <div style="margin-top:8px">
          <div class="input"><label>Target Class</label><select id="targetClass"><option value="">Select class</option><option>6</option><option>7</option><option>8</option><option>9</option><option>10</option><option>11</option><option>12</option></select></div>
          <div class="input"><label>Subject</label><input id="subject"></div>
          <div class="input"><label>Teacher Mobile (displayed to students)</label><input id="tmobile"></div>
          <div class="input"><label>Questions area (add up to 10). For each question add options and choose correct index.</label>
            <div id="questionsBlock"></div>
            <div style="margin-top:8px"><button class="btn" onclick="addQuestion()">Add Question</button></div>
          </div>
          <div style="margin-top:8px"><button class="btn" onclick="postQuiz()">Post Quiz</button></div>
        </div>
      </div>

      <div class="card" style="margin-top:14px">
```

```
      <h3 class="h2">Uploaded Quizzes</h3>
      <div id="uploadedQuizzes" class="small">No quizzes posted.</div>
    </div>
  </div>

  <div>
    <div class="card">
      <h3 class="h2">Upload Notes (PDF or text)</h3>
      <div class="input"><label>Target Class</label><select
id="noteClass"><option value="">Select
class</option><option>6</option><option>7</option><option>8</option><op
tion>9</option><option>10</option><option>11</option><option>12</option
></select></div>
      <div class="input"><label>Title</label><input id="noteTitle"></div>
      <div class="input"><label>Text Notes</label><textarea id="noteText"
rows="4"></textarea></div>
      <div class="input"><label>Upload PDF (choose file)</label><input
id="notePdf" type="file" accept="application/pdf"></div>
      <div style="margin-top:8px"><button class="btn"
onclick="uploadNote()">Upload Note</button></div>
    </div>

    <div class="card" style="margin-top:14px">
      <h3 class="h2">Track Results</h3>
      <div id="resultsTrack" class="small">No results yet.</div>
    </div>
  </div>
</div>
```

```
</div>


<script>

function getSession(){return
JSON.parse(localStorage.getItem('edu_session')||'null');}

function getUsers(){return JSON.parse(localStorage.getItem('edu_users')||'[]');}

function getQuizzes(){return
JSON.parse(localStorage.getItem('edu_quizzes')||'[]');}

function saveQuizzes(q){localStorage.setItem('edu_quizzes', JSON.stringify(q));}

function getNotes(){return
JSON.parse(localStorage.getItem('edu_notes')||'[]');}

function saveNotes(n){localStorage.setItem('edu_notes', JSON.stringify(n));}

function getResults(){return
JSON.parse(localStorage.getItem('edu_results')||'[]');}


const sess = getSession();

if(!sess || sess.role!=='teacher'){ alert('Teacher session not found. Please login
as teacher.'); location.href='login.html'; }

document.getElementById('userbox').innerHTML =
`<div><strong>${sess.name}</strong><div class="small">Assigned class: ${
(function(){ const u=getUsers().find(x=>x.email===sess.email); return u &&
u.assignedClass? u.assignedClass : 'Not set'; })() }</div></div>`;


document.getElementById('tmobile').value =
(getUsers().find(x=>x.email===sess.email)||{}).mobile || '';


let qCount=0;

function addQuestion(pref){

  if(qCount>=10){ alert('Maximum 10 questions'); return; }
```

```
  const idx=qCount;
  const block = document.getElementById('questionsBlock');
  const html = `<div id="q${idx}" style="border:1px dashed
#eee;padding:8px;border-radius:8px;margin-bottom:8px">
    <div><label>Q${idx+1}:</label><input style="width:100%" data-key="q"
placeholder="Question text"></div>
    <div style="display:flex;gap:8px;margin-top:6px">
      <input data-key="op0" placeholder="Option 1">
      <input data-key="op1" placeholder="Option 2">
    </div>
    <div style="display:flex;gap:8px;margin-top:6px">
      <input data-key="op2" placeholder="Option 3">
      <input data-key="op3" placeholder="Option 4">
    </div>
    <div style="margin-top:6px"><label>Correct option:</label>
      <select data-key="correct">
        <option value="">Select index (0-3)</option>
        <option value="0">0 (Option 1)</option>
        <option value="1">1 (Option 2)</option>
        <option value="2">2 (Option 3)</option>
        <option value="3">3 (Option 4)</option>
      </select>
    </div>
  </div>`;
  block.insertAdjacentHTML('beforeend', html);
  qCount++;
}
```

```
function postQuiz(){
  const target = document.getElementById('targetClass').value;
  const subj = document.getElementById('subject').value.trim();
  const tmobile = document.getElementById('tmobile').value.trim();
  if(!target || !subj){ alert('Select target class and enter subject'); return; }
  const questionsEls = Array.from(document.getElementById('questionsBlock').children);
  if(questionsEls.length===0){ alert('Add questions (up to 10).'); return; }
  if(questionsEls.length>10){ alert('Max 10 only'); return; }
  // build questions
  const qs = [];
  for(const el of questionsEls){
    const qtxt = el.querySelector('[data-key="q"]').value.trim();
    const ops = [el.querySelector('[data-key="op0"]').value.trim(), el.querySelector('[data-key="op1"]').value.trim(), el.querySelector('[data-key="op2"]').value.trim(), el.querySelector('[data-key="op3"]').value.trim()];
    const corr = el.querySelector('[data-key="correct"]').value;
    if(!qtxt || ops.some(o=>o==='') || corr===''){ alert('Fill all question fields and select correct option'); return; }
    qs.push({q:qtxt,options:ops,correct:parseInt(corr)});
  }
  const quizzes = getQuizzes();
  const id = 'quiz_' + Date.now();
  quizzes.push({id,subject:subj,teacherEmail:sess.email,teacherName:sess.name,teacherMobile:tmobile,targetClass:target,questions:qs,postedAt:new Date().toISOString()});
  saveQuizzes(quizzes);
```

```
  alert('Quiz posted for class '+target);
  // reset form
  document.getElementById('questionsBlock').innerHTML='';
  qCount=0;
  document.getElementById('subject').value='';
  document.getElementById('targetClass').value='';
  renderUploadedQuizzes();
}

function renderUploadedQuizzes(){
  const quizzes = getQuizzes().filter(q=> q.teacherEmail===sess.email);
  const area = document.getElementById('uploadedQuizzes');
  if(!quizzes.length){ area.innerHTML = 'No quizzes posted yet.'; return; }
  let out='<ol>';
  quizzes.forEach(q=>{
    out += `<li><strong>${q.subject}</strong> — Class ${q.targetClass} — Posted
${new Date(q.postedAt).toLocaleString()}</li>`;
  });
  out += '</ol>';
  area.innerHTML = out;
}

async function uploadNote(){
  const target = document.getElementById('noteClass').value;
  const title = document.getElementById('noteTitle').value.trim();
  const text = document.getElementById('noteText').value.trim();
  const file = document.getElementById('notePdf').files[0];
```

```
  if(!target || !title){ alert('Select target class and title'); return; }
  let pdfDataUrl = null;
  if(file){
    pdfDataUrl = await toDataURL(file);
  }
  const notes = getNotes();
  notes.push({id:'note_'+Date.now(),title,teacherEmail:sess.email,teacherName
:sess.name,targetClass:target,text,pdfDataUrl,postedAt:new
Date().toISOString()});
  saveNotes(notes);
  alert('Note uploaded for class '+target);
  document.getElementById('noteTitle').value='';
document.getElementById('noteText').value='';
document.getElementById('notePdf').value='';
}

function toDataURL(file){
  return new Promise((res,rej)=>{
    const r=new FileReader();
    r.onload=()=>res(r.result);
    r.onerror=()=>rej('err');
    r.readAsDataURL(file);
  });
}

function renderResultsTrack(){
  const res = getResults();
```

```
  const myClass = (getUsers().find(x=>x.email===sess.email)||{}).assignedClass;
  const area = document.getElementById('resultsTrack');
  const filtered = res.filter(r=> {
    // find student and check class
    const stu = getUsers().find(u=>u.email===r.studentEmail);
    return stu && stu.cls === myClass;
  });
  if(!filtered.length){ area.innerHTML = 'No student attempts yet.'; return; }
  // aggregate top per class
  filtered.sort((a,b)=> b.score - a.score || a.timeTaken - b.timeTaken);
  let out='<ol>';
  filtered.forEach(r=>{
    out += `<li><strong>${r.studentName}</strong> (${r.studentEmail}) — ${r.subject} — ${r.score}/10 — ${r.timeTaken}s — ${new Date(r.at).toLocaleString()}</li>`;
  });
  out += '</ol>';
  area.innerHTML = out;
}

// initial render
renderUploadedQuizzes();
renderResultsTrack();

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

games.html

```
<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width,initial-scale=1">

<title>EduBoard — Games</title>

<style>

body{font-family:Inter,system-ui,Segoe
UI,Roboto,Arial;margin:0;background:linear-
gradient(180deg,#fff,#f7fdff);padding:20px}

.header{display:flex;justify-content:space-between;align-items:center}

.container{max-width:1100px;margin:18px auto}

.card{background:white;padding:12px;border-radius:10px;box-shadow:0 8px
20px rgba(2,6,23,0.06);margin-bottom:12px}

.btn{padding:8px 12px;border-
radius:8px;border:none;background:#2563eb;color:white;cursor:pointer}

.small{font-size:13px;color:#556577}

.grid{display:flex;gap:18px;flex-wrap:wrap}

.game-tile{width:calc(20% - 14px);min-width:180px;padding:18px;border-
radius:12px;text-align:center;border:1px solid #eef6ff;background:linear-
gradient(180deg,#ffffff,#fbfdff)}

.locked{opacity:0.35}

.game-tile h4{margin:0 0 8px 0}

.note{font-size:14px;color:#334155;margin-top:8px}

</style>

</head>
```

```
<body>
<div class="header">
  <div><strong>EduBoard</strong> — Games</div>
  <div id="userbox"></div>
</div>
<div class="container">
  <div class="card">
    <div class="small">
      Games unlock based on <b>today’s quiz score</b>: 6→1 game, 7→2, 8→3, 9/10→all 5 games.
      After 24 hours, they will lock again until you attempt the new quiz.
    </div>
  </div>

  <div class="grid" id="gamesGrid"></div>
</div>

<script>
function getSession(){return JSON.parse(localStorage.getItem('edu_session')||'null');}
const sess = getSession();
if(!sess){ alert('Login required'); location.href='login.html'; }
document.getElementById('userbox').innerText = sess.name;

// get today's date (yyyy-mm-dd)
const today = new Date().toISOString().split('T')[0];
```

```
// stored quiz data
let lastQuiz = JSON.parse(localStorage.getItem('edu_quiz_record') || 'null');

let unlocked = 0;
if(lastQuiz && lastQuiz.date === today){
  unlocked = lastQuiz.score;
} else {
  unlocked = 0; // lock if quiz not taken today
}

// mapping score to games unlocked
const mapping = (s=>{ if(s>=9) return 5; if(s==8) return 3; if(s==7) return 2; if(s==6) return 1; return 0; })(unlocked);

const games = [
  {id:'g1',name:'Memory Pairs',desc:'Flip matching picture/letter pairs',file:'game_memory.html'},
  {id:'g2',name:'Sequence Recall',desc:'Remember & repeat sequences',file:'game_sequence.html'},
  {id:'g3',name:'Word Match',desc:'Match word pairs quickly',file:'game_wordmatch.html'},
  {id:'g4',name:'Sudoku',desc:'Classic number puzzle challenge',file:'game_sudoku.html'},
  {id:'g5',name:'Number Grid',desc:'Find numbers in order',file:'game_numbergrid.html'}
];

const grid=document.getElementById('gamesGrid');
```

```
games.forEach((g,idx)=>{
  const allowed = idx < mapping;
  const div = document.createElement('div');
  div.className='game-tile '+(allowed? '':'locked');
  div.innerHTML = `
    <h4>${g.name}</h4>
    <div class="small">${g.desc}</div>
    <div class="note">Today’s score: ${unlocked || 'No quiz yet'}</div>
    <div style="margin-top:12px">
      <button class="btn"
${allowed?`onclick="openGame('${g.file}')"`:'disabled'}>
        ${allowed? 'Play' : 'Locked'}
      </button>
    </div>`;
  grid.appendChild(div);
});

function openGame(file){ window.location.href = file; }
</script>
</body>
</html>
```

game_wordmatch.html

```
<!DOCTYPE html>
<html lang='en'>
```

```
<head>
  <meta charset='UTF-8'>
  <meta name='viewport' content='width=device-width, initial-scale=1.0'>
  <title>Word Match</title>
  <style>
    body{font-family:Arial;background:#f8fafc;text-align:center;}
    h1{color:#2563eb;}
    #game{display:grid;grid-template-columns:repeat(4,120px);grid-
gap:10px;justify-content:center;margin-top:20px;}
    .card{background:#2563eb;color:#fff;padding:20px;border-
radius:8px;cursor:pointer;}
    .flipped{background:#fff;color:#000;}
  </style>
</head>
<body>
  <h1>Word Match</h1>
  <p>Match synonyms!</p>
  <div id="game"></div>
  <script>
    const
pool=[['big','large'],['small','tiny'],['happy','joyful'],['fast','quick'],['cold','chilly'],['
smart','clever'],['angry','mad'],['rich','wealthy'],['strong','powerful'],['safe','secu
re'],['bright','luminous'],['dark','dim']];
    let chosen=pool.sort(()=>0.5-Math.random()).slice(0,6);
    let words=[];chosen.forEach(p=>words.push(...p));
    let deck=words.sort(()=>0.5-Math.random());
    const game=document.getElementById('game');
    let flipped=[],matched=[];
```

```
  function findPair(w){return chosen.find(p=>p.includes(w));}
  deck.forEach(w=>{let d=document.createElement('div');d.className='card';d.textContent=w;d.onclick=function(){if(flipped.length<2&&!flipped.includes(this)&&!matched.includes(this)){this.classList.add('flipped');flipped.push(this);if(flipped.length===2){let p1=findPair(flipped[0].textContent);let p2=findPair(flipped[1].textContent);if(p1===p2){matched.push(...flipped);flipped=[];if(matched.length===deck.length)alert('All matched!');}else{setTimeout(()=>{flipped.forEach(c=>c.classList.remove('flipped'));flipped=[];},1000);}}}};game.appendChild(d);});
 </script>
</body>
</html>
```

game_sudoku.html

```
<!DOCTYPE html>
<html lang='en'>
<head>
 <meta charset='UTF-8'>
 <meta name='viewport' content='width=device-width, initial-scale=1.0'>
 <title>Sudoku</title>
 <style>
  body{font-family:Arial;background:#f8fafc;text-align:center;}
  h1{color:#2563eb;}
  table{border-collapse:collapse;margin:20px auto;}
  td{border:1px solid #2563eb;width:40px;height:40px;text-align:center;font-size:18px;}
```

```
    input{width:38px;height:38px;text-align:center;font-size:18px;border:none;}
    .fixed{background:#e0f2fe;}
  </style>
</head>
<body>
  <h1>Sudoku</h1>
  <p>Select difficulty:</p>
  <select id="difficulty">
    <option value="easy">Easy (20s preview)</option>
    <option value="medium">Medium (20s preview)</option>
    <option value="hard">Hard (10s preview)</option>
  </select>
  <button onclick="startSudoku()">Start</button>
  <table id="board"></table>
  <button onclick="checkSudoku()">Check Solution</button>
  <script>
    const board=document.getElementById('board');
    let solution=[];
    function generateFullBoard(){let
arr=Array.from({length:9},()=>Array(9).fill(0));function canPlace(r,c,num){for(let
i=0;i<9;i++){if(arr[r][i]===num||arr[i][c]===num)return false;}let
sr=Math.floor(r/3)*3,sc=Math.floor(c/3)*3;for(let i=0;i<3;i++)for(let
j=0;j<3;j++)if(arr[sr+i][sc+j]===num)return false;return true;}function
solve(pos=0){if(pos===81)return true;let
r=Math.floor(pos/9),c=pos%9;if(arr[r][c]!==0)return solve(pos+1);let
nums=[1,2,3,4,5,6,7,8,9].sort(()=>0.5-Math.random());for(let num of
nums){if(canPlace(r,c,num)){arr[r][c]=num;if(solve(pos+1))return
true;arr[r][c]=0;}}return false;}solve();return arr;}
```

```
  function startSudoku(){board.innerHTML='';solution=generateFullBoard();let diff=document.getElementById('difficulty').value;let blanks=30;if(diff==='medium')blanks=40;if(diff==='hard')blanks=50;let puzzle=solution.map(r=>[...r]);for(let i=0;i<blanks;i++){let r=Math.floor(Math.random()*9),c=Math.floor(Math.random()*9);puzzle[r][c]=0;}for(let r=0;r<9;r++){let row=document.createElement('tr');for(let c=0;c<9;c++){let cell=document.createElement('td');if(puzzle[r][c]!==0){cell.innerHTML='<input value="'+puzzle[r][c]+'" readonly class="fixed">';}else{cell.innerHTML='<input>';}row.appendChild(cell);}board.appendChild(row);}}

  function checkSudoku(){let rows=board.getElementsByTagName('tr');for(let r=0;r<9;r++){let cells=rows[r].getElementsByTagName('td');for(let c=0;c<9;c++){let val=cells[c].firstChild.value;if(Number(val)!==solution[r][c]){alert('Incorrect solution!');return;}}}alert('Correct! Well done!');}

 </script>

</body>

</html>
```

game_sequence.html

```
<!DOCTYPE html>

<html lang='en'>

<head>

 <meta charset='UTF-8'>

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

 <title>Sequence Recall</title>

 <style>

  body{font-family:Arial;background:#f8fafc;text-align:center;}
```

```
   h1{color:#2563eb;}
   #grid{display:grid;grid-template-columns:repeat(3,100px);grid-gap:10px;justify-content:center;margin-top:20px;}
   .cell{width:100px;height:100px;background:#2563eb;border-radius:8px;cursor:pointer;}
   .active{background:#facc15!important;}
  </style>
</head>
<body>
  <h1>Sequence Recall</h1>
  <p>Watch the sequence, then repeat!</p>
  <button onclick="startGame()">Start</button>
  <div id="grid"></div>
  <script>
   const grid=document.getElementById('grid');
   let cells=[]; let sequence=[]; let player=[];
   for(let i=0;i<9;i++){let d=document.createElement('div');d.className='cell';d.dataset.index=i;grid.appendChild(d);cells.push(d);}
   function startGame(){sequence=[];player=[];nextRound();}
   function nextRound(){player=[];sequence.push(Math.floor(Math.random()*9));showSequence();}
   function showSequence(){let i=0;let interval=setInterval(()=>{let idx=sequence[i];cells[idx].classList.add('active');setTimeout(()=>cells[idx].classList.remove('active'),500);i++;if(i>=sequence.length){clearInterval(interval);enableInput();}},1000);}
   function enableInput(){cells.forEach(c=>c.onclick=()=>{player.push(Number(c.dataset.index));check();});}
```

```
  function check(){for(let i=0;i<player.length;i++){if(player[i]!==sequence[i]){alert('Wrong! Final score: '+(sequence.length-1));return;}}if(player.length===sequence.length){setTimeout(nextRound,1000);}}

 </script>

</body>

</html>
```

game_numbergrid.html

```
<!DOCTYPE html>

<html lang='en'>

<head>

 <meta charset='UTF-8'>

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

 <title>Number Grid</title>

 <style>

  body{font-family:Arial;background:#f8fafc;text-align:center;}

  h1{color:#2563eb;}

  #grid{display:grid;grid-template-columns:repeat(5,60px);grid-gap:5px;justify-content:center;margin-top:20px;}

  .cell{width:60px;height:60px;background:#2563eb;color:#fff;font-size:20px;display:flex;align-items:center;justify-content:center;border-radius:6px;cursor:pointer;}

 </style>

</head>

<body>
```

```
<h1>Number Grid</h1>
<p>Select difficulty and click numbers in order!</p>
<select id="difficulty">
<option value="easy">Easy (1-25, 20s preview)</option>
<option value="medium">Medium (1-36, 20s preview)</option>
<option value="hard">Hard (1-49, 10s preview)</option>
</select>
<button onclick="startGame()">Start</button>
<div id="grid"></div>
<script>
const grid=document.getElementById('grid');
let expected=1,numbers=[],previewTime=5000;
function startGame(){
grid.innerHTML='';expected=1;
let diff=document.getElementById('difficulty').value;
let max=25,preview=20000;
if(diff==='medium'){max=36;preview=20000;}
if(diff==='hard'){max=49;preview=10000;}
numbers=[...Array(max).keys()].map(x=>x+1).sort(()=>0.5-Math.random());
numbers.forEach(n=>{let
d=document.createElement('div');d.className='cell';d.textContent=n;grid.app
endChild(d);});
setTimeout(()=>{Array.from(grid.children).forEach(c=>c.textContent='?');ena
bleClick();},preview);
}
function enableClick(){
Array.from(grid.children).forEach(d=>{
```

```
      d.onclick=()=>{let
n=Number(d.dataset.num||d.textContent);if(d.textContent==='?')return;
        n=Number(d.textContent)||n;
        if(n===expected){d.textContent='✓';expected++;if(expected>numbers.len
gth)alert('Completed!');}else alert('Wrong!');};
     });
   }
 </script>
</body>
</html>
```

game_memory.html

```
<!DOCTYPE html>
<html lang='en'>
<head>
 <meta charset='UTF-8'>
 <meta name='viewport' content='width=device-width, initial-scale=1.0'>
 <title>Memory Pairs</title>
 <style>
   body { font-family: Arial, sans-serif; background:#f8fafc; text-align:center; }
   h1 { color:#2563eb; }
   #game { display:grid; grid-template-columns: repeat(4, 100px); grid-
gap:10px; justify-content:center; margin-top:20px; }
   .card { width:100px; height:100px; background:#2563eb; color:white; font-
size:2em; display:flex; align-items:center; justify-content:center; cursor:pointer;
border-radius:10px; }
```

```
    .flipped { background:#fff; color:#000; cursor:default; }
  </style>
</head>
<body>
  <h1>Memory Pairs</h1>
  <p>Match all pairs!</p>
  <div id='game'></div>
  <script>
    const symbols =
['😀','🎵','🍉','🚗','🐶','⚽','🌟','📋','🎲','💡','🎨','✈️','🍩','🔑','🏀','🎤'];
    let chosen = symbols.sort(() => 0.5 - Math.random()).slice(0,8);
    let deck = [...chosen, ...chosen].sort(() => 0.5 - Math.random());
    const game = document.getElementById('game');
    let flipped = [];
    let matched = [];
    deck.forEach((sym,i)=>{
      let card=document.createElement('div');
      card.className='card';
      card.dataset.symbol=sym;
      card.dataset.index=i;
      card.textContent='?';
      card.onclick=function(){
        if(flipped.length<2 && !flipped.includes(this) && !matched.includes(this)){
          this.textContent=this.dataset.symbol;
          this.classList.add('flipped');
          flipped.push(this);
```

```
        if(flipped.length===2){
          if(flipped[0].dataset.symbol===flipped[1].dataset.symbol){
            matched.push(...flipped);
            flipped=[];
            if(matched.length===deck.length){ alert('You matched all!'); }
          } else {
            setTimeout(()=>{
              flipped.forEach(c=>{c.textContent='?';c.classList.remove('flipped');});
              flipped=[];
            },1000);
          }
        }
      }
      game.appendChild(card);
    });
  </script>
</body>
</html>
```