```
import React, { useState } from 'react';
import {
  Box,
  Typography,
  Input,
  TextField,
  Button,
  Card,
  CardContent,
} from '@mui/material';

interface Question {
  id: number;
  question: string;
  options: string[];
  correctAnswer: string;
}

interface TeacherPanelProps {
  onSave: (questions: Question[], startTime: string, duration: number) => void;
}

const TeacherPanel: React.FC<TeacherPanelProps> = ({ onSave }) => {
  const [questions, setQuestions] = useState<Question[]>([]);
  const [startTime, setStartTime] = useState('');
  const [duration, setDuration] = useState(30);

  const handleFileUpload = (e: React.ChangeEvent<HTMLInputElement>) => {
    const file = e.target.files?.[0];
    if (!file) return;
    const reader = new FileReader();
    reader.onload = (event) => {
      try {
        const json = JSON.parse(event.target?.result as string);
        if (Array.isArray(json)) {
          setQuestions(json);
        } else {
          alert('Invalid JSON format.');
        }
      } catch {
        alert('Failed to read JSON file.');
      }
    };
    reader.readAsText(file);
  };

  const handleQuestionEdit = (index: number, key: keyof Question, value: string) => {
    const updated = [...questions];
  if (key === 'options') return; // Options are handled separately
  updated[index][key] = value;
  setQuestions(updated);
};

const handleOptionEdit = (qIndex: number, oIndex: number, value: string) => {
  const updated = [...questions];
  updated[qIndex].options[oIndex] = value;
  setQuestions(updated);
};

return (
  <Box sx={{ p: 3 }}>
    <Typography variant="h5">Teacher Panel - Upload & Schedule</Typography>

    <Box sx={{ mt: 2 }}>
      <Typography>Upload Graded Exercise File (.json)</Typography>
      <Input type="file" inputProps={{ accept: '.json' }} onChange={handleFileUpload} />
    </Box>

    <Box sx={{ my: 2 }}>
      <TextField
        type="datetime-local"
        label="Start Time"
        value={startTime}
        onChange={(e) => setStartTime(e.target.value)}
        InputLabelProps={{ shrink: true }}
        sx={{ mr: 2 }}
      />
      <TextField
        label="Duration (mins)"
        type="number"
        value={duration}
        onChange={(e) => setDuration(Number(e.target.value))}
      />
    </Box>

    {questions.map((q, i) => (
      <Card key={q.id} sx={{ my: 2 }}>
        <CardContent>
          <TextField
            fullWidth
            label={`Question ${i + 1}`}
            value={q.question}
            onChange={(e) => handleQuestionEdit(i, 'question', e.target.value)}
          />
          {q.options.map((opt, j) => (
            <TextField
```

```
                key={j}
                fullWidth
                label={`Option ${j + 1}`}
                value={opt}
                onChange={(e) => handleOptionEdit(i, j, e.target.value)}
                sx={{ mt: 1 }}
              />
            ))}
            <TextField
              fullWidth
              label="Correct Answer"
              value={q.correctAnswer}
              onChange={(e) => handleQuestionEdit(i, 'correctAnswer', e.target.value)}
              sx={{ mt: 2 }}
            />
          </CardContent>
        </Card>
      ))}

      <Button
        variant="contained"
        onClick={() => onSave(questions, startTime, duration)}
        sx={{ mt: 2 }}
      >
        Save Exam
      </Button>
    </Box>
  );
};

export default TeacherPanel;
```

Part 2

```
import React, { useEffect, useState } from 'react';
import {
  Box,
  Typography,
  Card,
  CardContent,
  Radio,
  RadioGroup,
  FormControlLabel,
  Button,
} from '@mui/material';
import dayjs from 'dayjs';

interface Question {
  id: number;
  question: string;
  options: string[];
  correctAnswer: string;
}

interface StudentExamProps {
  questions: Question[];
  startTime: string;
  duration: number;
}

const StudentExam: React.FC<StudentExamProps> = ({ questions, startTime, duration }) =>
{
  const [answers, setAnswers] = useState<Record<number, string>>({});
  const [score, setScore] = useState<number | null>(null);
  const [now, setNow] = useState<string>(dayjs().toISOString());

  useEffect(() => {
    const interval = setInterval(() => setNow(dayjs().toISOString()), 1000);
    return () => clearInterval(interval);
  }, []);

  const isAvailable = () => {
    const start = dayjs(startTime);
    const end = start.add(duration, 'minute');
    return dayjs(now).isAfter(start) && dayjs(now).isBefore(end);
  };

  const isExpired = () => {
    const end = dayjs(startTime).add(duration, 'minute');
    return dayjs(now).isAfter(end);
  };

  const handleAnswerChange = (id: number, value: string) => {
    setAnswers((prev) => ({ ...prev, [id]: value }));
  };

  const handleSubmit = () => {
    let score = 0;
    questions.forEach((q) => {
      if (answers[q.id] === q.correctAnswer) score++;
    });
    setScore(score);
  };

  if (!startTime || !duration || questions.length === 0)
    return <Typography>Please wait for the teacher to configure the exam.</Typography>;

  if (!isAvailable())
    return (
      <Typography color="warning.main">
        {isExpired() ? 'Exam has ended.' : 'Exam is not yet available.'}
      </Typography>
    );

  return (
    <Box sx={{ p: 3 }}>
      <Typography variant="h5">Student Exam</Typography>
      {questions.map((q) => (
        <Card key={q.id} sx={{ my: 2 }}>
          <CardContent>
            <Typography>{q.question}</Typography>
            <RadioGroup
              value={answers[q.id] || ''}
              onChange={(e) => handleAnswerChange(q.id, e.target.value)}
            >
              {q.options.map((opt) => (
                <FormControlLabel key={opt} value={opt} control={<Radio />} label={opt} />
              ))}
            </RadioGroup>
          </CardContent>
        </Card>
      ))}
      <Button variant="contained" onClick={handleSubmit}>
        Submit
      </Button>
      {score !== null && (
        <Typography variant="h6" sx={{ mt: 2 }}>
          Score: {score} / {questions.length}
        </Typography>
      )}
    </Box>
  );
};

export default StudentExam;
```

Part 3

```
import React, { useState } from 'react';
import TeacherPanel from './TeacherPanel';
import StudentExam from './StudentExam';
import { Box, Divider } from '@mui/material';

interface Question {
  id: number;
  question: string;
  options: string[];
  correctAnswer: string;
}

const App: React.FC = () => {
  const [questions, setQuestions] = useState<Question[]>([]);
  const [startTime, setStartTime] = useState('');
  const [duration, setDuration] = useState(30);

  const handleSave = (q: Question[], start: string, dur: number) => {
    setQuestions(q);
    setStartTime(start);
    setDuration(dur);
  };

  return (
    <Box>
      <TeacherPanel onSave={handleSave} />
      <Divider sx={{ my: 3 }} />
      <StudentExam questions={questions} startTime={startTime} duration={duration} />
    </Box>
  );
};

export default App;
```