

## Add New Course

Title:

Description:

Fees:

Course Notes (PDF):

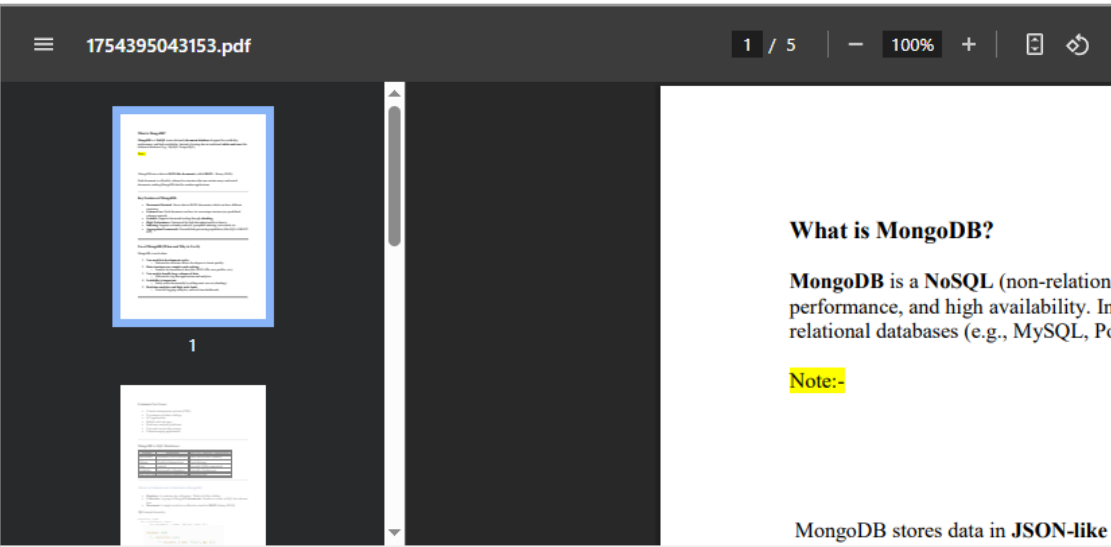
No file chosen

### Lectures

No file chosen

← → ↻ ⓘ localhost:3000/courses

☰ 1754395043153.pdf 1 / 5 | - 100% + | 📄 ↻



**What is MongoDB?**

MongoDB is a NoSQL (non-relational, performance, and high availability). In relational databases (e.g., MySQL, PostgreSQL)

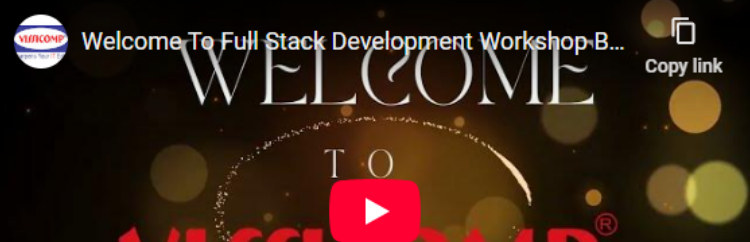
**Note:-**

MongoDB stores data in JSON-like

**Lectures**

variables - 2

**Lecture Video:**



## ✓ 1. app.js – Full Backend with Course + Lecture PDF Upload Support

```
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const multer = require('multer');
const path = require('path');

const app = express();

// Connect to MongoDB
mongoose.connect('mongodb://localhost:27017/mongo-crud', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});

// Define Course schema with embedded lectures
const courseSchema = new mongoose.Schema({
  title: String,
```

```

    description: String,
    fees: Number,
    notes: String, // course-level notes PDF
    lectures: [
      {
        title: String,
        duration: String,
        videoUrl: String,
        notes: String // lecture-specific PDF
      }
    ]
  });
const Course = mongoose.model('Course', courseSchema);

// Multer setup
const storage = multer.diskStorage({
  destination: (req, file, cb) => {
    cb(null, 'uploads/');
  },
  filename: (req, file, cb) => {
    cb(null, Date.now() + path.extname(file.originalname));
  }
});
const upload = multer({ storage });

const cpUpload = upload.fields([
  { name: 'notes', maxCount: 1 },
  { name: 'lectureNotes', maxCount: 20 }
]);

// Middleware
app.set('view engine', 'ejs');
app.use(bodyParser.urlencoded({ extended: true }));
app.use(express.static(path.join(__dirname, 'uploads')));

// Routes
app.get('/', (req, res) => res.redirect('/courses'));

app.get('/courses', async (req, res) => {
  const courses = await Course.find();
  res.render('index', { courses });
});

app.get('/courses/new', (req, res) => {
  res.render('new');
});

app.post('/courses', cpUpload, async (req, res) => {
  const { title, description, fees } = req.body;
  const files = req.files || {};
  const mainNotes = files.notes?.[0]?.filename || null;
  const lectureNoteFiles = files.lectureNotes || [];

  let lectures = [];

  if (Array.isArray(req.body['lecture-title'])) {
    lectures = req.body['lecture-title'].map((_, i) => ({
      title: req.body['lecture-title'][i],
      duration: req.body['lecture-duration'][i],
      videoUrl: req.body['lecture-videoUrl'][i],
    }));
  }
});

```

```

        notes: lectureNoteFiles[i] ? lectureNoteFiles[i].filename :
null
    ));
} else if (req.body['lecture-title']) {
    lectures = [{
        title: req.body['lecture-title'],
        duration: req.body['lecture-duration'],
        videoUrl: req.body['lecture-videoUrl'],
        notes: lectureNoteFiles[0]?.filename || null
    }];
}

    await Course.create({ title, description, fees, notes: mainNotes,
lectures });
    res.redirect('/courses');
});

app.get('/courses/:id/edit', async (req, res) => {
    const course = await Course.findById(req.params.id);
    res.render('edit', { course });
});

app.post('/courses/:id', cpUpload, async (req, res) => {
    const { title, description, fees } = req.body;
    const files = req.files || {};
    const mainNotes = files.notes?.[0]?.filename;
    const lectureNoteFiles = files.lectureNotes || [];

    let lectures = [];

    if (Array.isArray(req.body['lecture-title'])) {
        lectures = req.body['lecture-title'].map((_, i) => ({
            title: req.body['lecture-title'][i],
            duration: req.body['lecture-duration'][i],
            videoUrl: req.body['lecture-videoUrl'][i],
            notes: lectureNoteFiles[i] ? lectureNoteFiles[i].filename :
null
        }));
    } else if (req.body['lecture-title']) {
        lectures = [{
            title: req.body['lecture-title'],
            duration: req.body['lecture-duration'],
            videoUrl: req.body['lecture-videoUrl'],
            notes: lectureNoteFiles[0]?.filename || null
        }];
    }

    const updateData = {
        title,
        description,
        fees,
        lectures
    };

    if (mainNotes) {
        updateData.notes = mainNotes;
    }

    await Course.findByIdAndUpdate(req.params.id, updateData);
    res.redirect('/courses');
});

```

```

app.post('/courses/:id/delete', async (req, res) => {
  await Course.findByIdAndDelete(req.params.id);
  res.redirect('/courses');
});

// Start server
const PORT = 3000;
app.listen(PORT, () => {
  console.log(`Server running at http://localhost:${PORT}`);
});

```

---

## ✓ 2. views/index.ejs

```

<!DOCTYPE html>
<html>
<head>
  <title>Courses</title>
</head>
<body>
  <h1>All Courses</h1>
  <a href="/courses/new">Add New Course</a>

  <ul>
    <% courses.forEach(course => { %>
      <li style="border:1px solid #ccc; padding:10px; margin:10px;">
        <h2><%= course.title %></h2>
        <p><strong>Description:</strong> <%= course.description
%></p>
        <p><strong>Fees:</strong> ₹<%= course.fees %></p>

        <% if (course.notes) { %>
          <h4>Course Notes (PDF)</h4>
          <iframe src="/<%= course.notes %>" width="100%"
height="400px"></iframe>
          <% } %>

        <% if (course.lectures && course.lectures.length) { %>
          <h3>Lectures</h3>
          <% course.lectures.forEach(lec => { %>
            <div style="margin-bottom: 20px;">
              <strong><%= lec.title %></strong> - <%=
lec.duration %><br>

              <% if (lec.videoUrl) { %>
                <h4>Lecture Video:</h4>
                <iframe width="560" height="315"
src="<%= lec.videoUrl %>"
frameborder="0"
allow="accelerometer; autoplay; clipboard-write; encrypted-media;
gyroscope; picture-in-picture"
allowfullscreen>
                </iframe>
              <% } %>

              <% if (lec.notes) { %>

```

```

                                <h4>Lecture Notes (PDF):</h4>
                                <iframe src="/<%= lec.notes %>"
width="100%" height="300px"></iframe>
                                <% } %>
                            </div>
                        <% }) %>
                    <% } %>

                <a href="/courses/<%= course._id %>/edit">Edit</a>
                <form method="POST" action="/courses/<%= course._id
%>/delete" style="display:inline;">
                    <button type="submit" onclick="return confirm('Delete
this course?')">Delete</button>
                </form>
            </li>
        <% }) %>
    </ul>
</body>
</html>

```

---

### ✓ 3. views/new.ejs

```

<!DOCTYPE html>
<html>
<head>
    <title>New Course</title>
</head>
<body>
    <h1>Add New Course</h1>
    <form action="/courses" method="POST" enctype="multipart/form-data">
        <label>Title:</label><br>
        <input type="text" name="title" required><br><br>

        <label>Description:</label><br>
        <textarea name="description" required></textarea><br><br>

        <label>Fees:</label><br>
        <input type="number" name="fees" required><br><br>

        <label>Course Notes (PDF):</label><br>
        <input type="file" name="notes" accept="application/pdf"><br><br>

        <h3>Lectures</h3>
        <div id="lectures-container">
            <div class="lecture">
                <input type="text" name="lecture-title"
placeholder="Lecture Title" required>
                <input type="text" name="lecture-duration"
placeholder="Duration">
                <input type="text" name="lecture-videoUrl"
placeholder="Video URL">
                <input type="file" name="lectureNotes"
accept="application/pdf">
            </div>
            <div>
                <button type="button" onclick="addLecture()">+ Add
Lecture</button><br><br>

```

```

        <button type="submit">Create Course</button>
    </form>

    <script>
        function addLecture() {
            const container = document.getElementById('lectures-
container');
            const div = document.createElement('div');
            div.classList.add('lecture');
            div.innerHTML = `
                <input type="text" name="lecture-title"
placeholder="Lecture Title" required>
                <input type="text" name="lecture-duration"
placeholder="Duration">
                <input type="text" name="lecture-videoUrl"
placeholder="Video URL">
                <input type="file" name="lectureNotes"
accept="application/pdf">
            `;
            container.appendChild(div);
        }
    </script>
</body>
</html>

```

---



#### 4. views/edit.ejs

(Similar to new.ejs, but with values pre-filled)

```

<!DOCTYPE html>
<html>
<head>
    <title>Edit Course</title>
</head>
<body>
    <h1>Edit Course</h1>
    <form action="/courses/<%= course._id %>" method="POST"
enctype="multipart/form-data">
        <input type="text" name="title" value="<%= course.title %>"
required><br><br>
        <textarea name="description"><%= course.description
%></textarea><br><br>
        <input type="number" name="fees" value="<%= course.fees %>"
required><br><br>

        <% if (course.notes) { %>
            <p>Current Notes: <a href="/<%= course.notes %>"
target="_blank">Download</a></p>
        <% } %>
        <label>Upload New Course PDF:</label>
        <input type="file" name="notes" accept="application/pdf"><br><br>

    <h3>Lectures</h3>
    <div id="lectures-container">
        <% course.lectures.forEach(lec => { %>
            <div class="lecture">

```

```

        <input type="text" name="lecture-title" value="<%=
lec.title %>" required>
        <input type="text" name="lecture-duration" value="<%=
lec.duration %>">
        <input type="text" name="lecture-videoUrl" value="<%=
lec.videoUrl %>">
        <input type="file" name="lectureNotes"
accept="application/pdf">
        <% if (lec.notes) { %>
            <a href="/<%= lec.notes %>" target="_blank">Current
PDF</a>
        <% } %>
    </div>
    <% }) %>
</div>
    <button type="button" onclick="addLecture()">+ Add
Lecture</button><br><br>

    <button type="submit">Update Course</button>
</form>

<script>
    function addLecture() {
        const container = document.getElementById('lectures-
container');
        const div = document.createElement('div');
        div.classList.add('lecture');
        div.innerHTML = `
            <input type="text" name="lecture-title"
placeholder="Lecture Title" required>
            <input type="text" name="lecture-duration"
placeholder="Duration">
            <input type="text" name="lecture-videoUrl"
placeholder="Video URL">
            <input type="file" name="lectureNotes"
accept="application/pdf">
        `;
        container.appendChild(div);
    }
</script>
</body>
</html>

```