import com.itextpdf.text.\*;

import com.itextpdf.text.pdf.PdfWriter;

import java.io.FileOutputStream;

public class ToDoListApiPdf {

public static void main(String[] args) {

Document document = new Document();

try {

PdfWriter.getInstance(document, new FileOutputStream("ToDoList\_API\_Planning.pdf"));

document.open();

Font titleFont = FontFactory.getFont(FontFactory.HELVETICA\_BOLD, 18);

Font sectionFont = FontFactory.getFont(FontFactory.HELVETICA\_BOLD, 14);

Font normalFont = FontFactory.getFont(FontFactory.HELVETICA, 12);

document.add(new Paragraph("Assignment 7: API Planning for To-Do List App", titleFont));

document.add(Chunk.NEWLINE);

document.add(new Paragraph("🎯 Objective", sectionFont));

document.add(new Paragraph("Identify and plan the necessary RESTful APIs required to support the full CRUD functionality of a To-Do List application built with React.js.", normalFont));

document.add(new Paragraph("\n📚 Overview: CRUD Operations", sectionFont));

document.add(new Paragraph("- Create: Add a new task using POST\n- Read: Get all tasks or a specific task using GET\n- Update: Modify task details using PUT/PATCH\n- Delete: Remove a task using DELETE", normalFont));

document.add(new Paragraph("\n📌 1. Add Task", sectionFont));

document.add(new Paragraph("Endpoint: POST /api/tasks\nDescription: Creates a new task.\nRequest Body: { title, description, dueDate, priority }\nResponse: { id, title, description, dueDate, priority, completed, createdAt }", normalFont));

document.add(new Paragraph("\n📌 2. Get All Tasks", sectionFont));

document.add(new Paragraph("Endpoint: GET /api/tasks\nDescription: Fetch all tasks.\nResponse: [ { id, title, completed }, ... ]", normalFont));

document.add(new Paragraph("\n📌 3. Get Single Task", sectionFont));

document.add(new Paragraph("Endpoint: GET /api/tasks/:id\nDescription: Get details of a specific task.\nResponse: { id, title, description, dueDate, priority, completed, createdAt }", normalFont));

document.add(new Paragraph("\n📌 4. Update Task", sectionFont));

document.add(new Paragraph("Endpoint: PUT /api/tasks/:id\nRequest Body: { title, description, dueDate, priority }\nResponse: { message: Task updated successfully }", normalFont));

document.add(new Paragraph("\n📌 5. Toggle Task Completion", sectionFont));

document.add(new Paragraph("Endpoint: PATCH /api/tasks/:id/complete\nRequest Body: { completed: true/false }\nResponse: { message: Task marked as completed }", normalFont));

document.add(new Paragraph("\n📌 6. Delete Task", sectionFont));

document.add(new Paragraph("Endpoint: DELETE /api/tasks/:id\nResponse: { message: Task deleted successfully }", normalFont));

document.add(new Paragraph("\n🔁 Summary of API Endpoints", sectionFont));

document.add(new Paragraph(

"POST /api/tasks - Add new task\n" +

"GET /api/tasks - Get all tasks\n" +

"GET /api/tasks/:id - Get one task\n" +

"PUT /api/tasks/:id - Update task\n" +

"PATCH /api/tasks/:id/complete - Toggle complete\n" +

"DELETE /api/tasks/:id - Delete task", normalFont));

document.add(new Paragraph("\n🧠 Reasoning Behind API Design", sectionFont));

document.add(new Paragraph("- RESTful structure ensures scalability\n- Use of UUIDs, timestamps, JSON responses\n- Clear endpoint naming", normalFont));

document.add(new Paragraph("\n⚠️ Potential Challenges", sectionFont));

document.add(new Paragraph("- ID uniqueness\n- Input validation\n- Error handling\n- Syncing state with frontend\n- Future auth support", normalFont));

document.add(new Paragraph("\n✅ CRUD Support Overview", sectionFont));

document.add(new Paragraph("Create -> POST /api/tasks\nRead -> GET /api/tasks, GET /api/tasks/:id\nUpdate -> PUT/PATCH /api/tasks/:id\nDelete -> DELETE /api/tasks/:id", normalFont));

document.close();

System.out.println()

} catch (Exception e) {

e.printStackTrace();

}

}

}