Execution of PDF Generation Code

Below is the line-by-line explanation of the provided C# code for generating a PDF report using iTextSharp.

# Code Execution Explanation

## DataTable dt = getData("PR\_TBLSALARYMST\_SelectAll");

Fetches data from the database using the stored procedure 'PR\_TBLSALARYMST\_SelectAll'. The data is returned in a DataTable (dt) containing ID, EMPID, MONTH, and SALARY columns.

## var document = new Document();

Creates a new iTextSharp Document object which represents the PDF being built.

## using (var memoryStream = new MemoryStream())

Initializes a MemoryStream to temporarily hold the PDF content in memory.

## PdfWriter.GetInstance(document, memoryStream);

Binds the PDF document to the memory stream so that all written content goes into it.

## document.Open();

Opens the document for writing content like paragraphs and tables.

## var titleFont = FontFactory.GetFont(FontFactory.HELVETICA\_BOLD, 18);

Creates a font style (Helvetica Bold, size 18) for the PDF title.

## document.Add(new Paragraph("Product List", titleFont));

Adds the title 'Product List' to the PDF using the bold font.

## document.Add(new Paragraph(" "));

Adds an empty paragraph for spacing.

## var table = new PdfPTable(4);

Creates a table with 4 columns (for ID, EMPID, MONTH, SALARY).

## table.AddCell("ID"); ...

Adds header cells for ID, EMPID, MONTH, and SALARY.

## foreach (DataRow row in dt.Rows) { ... }

Loops through each row in the DataTable and extracts ID, EMPID, MONTH, and SALARY values. Each value is converted to string and added as a cell in the table.

## document.Add(table);

Adds the completed table into the PDF document.

## document.Close();

Finalizes and closes the document, preventing further edits.

## var bytes = memoryStream.ToArray();

Converts the in-memory PDF content into a byte array.

## return File(bytes, "application/pdf", "SalaryReport.pdf");

Returns the byte array as a downloadable PDF file with the name 'SalaryReport.pdf'.

Summary:

This code retrieves salary data from the database and generates a PDF report in memory using iTextSharp. The PDF includes a title and a table listing employee salary details. Finally, it returns the PDF as a downloadable file to the client.