ExcelGeneratorClass.java

```java
package in.gov.icegate.igstUtil.service;

import java.io.*;
import java.sql.*;

import org.apache.poi.ss.usermodel.*;
import org.apache.poi.xssf.usermodel.*;

import in.gov.icegate.common.db.DBConnection;


public class ExcelGeneratorClass {

    public static void main(String[] args) throws ClassNotFoundException {
        System.out.println("inside main");
        ExcelGeneratorClass simpleDb2ExcelExporter = new ExcelGeneratorClass();
        simpleDb2ExcelExporter.export();
    }


    public void export() throws ClassNotFoundException {


        Connection connection = null;
        PreparedStatement statement = null;
        ResultSet result = null;

        String excelFilePath = "D:\\Corona-Help-Database.xlsx";

        try {

            System.out.println("inside export");

            connection = DBConnection.getConnection();

            statement = connection.prepareStatement("SELECT * FROM CORONA_TRADE_HELP");

            result = statement.executeQuery();

            XSSFWorkbook workbook = new XSSFWorkbook();
            XSSFSheet sheet = workbook.createSheet("Corona-Help-Database");

            writeHeaderLine(sheet);

            writeDataLines(result, workbook, sheet);

            FileOutputStream outputStream = new FileOutputStream(excelFilePath);
```

Writable

ExcelGeneratorClass.java

```
47              FileOutputStream outputStream = new FileOutputStream(excelFilePath);
48              workbook.write(outputStream);
49              workbook.close();
50
51              statement.close();
52
53          } catch (SQLException e) {
54              System.out.println("Datababse error:");
55              e.printStackTrace();
56          } catch (IOException e) {
57              System.out.println("File IO error:");
58              e.printStackTrace();
59          }
60      }
61
62      private void writeHeaderLine(XSSFSheet sheet) {
63
64          System.out.println("inside writeHeaderLine");
65
66          Row headerRow = sheet.createRow(0);
67
68          Cell headerCell = headerRow.createCell(0);
69          headerCell.setCellValue("name");
70
71          headerCell = headerRow.createCell(1);
72          headerCell.setCellValue("iec");
73
74          headerCell = headerRow.createCell(2);
75          headerCell.setCellValue("email");
76
77          headerCell = headerRow.createCell(3);
78          headerCell.setCellValue("mobile");
79
80          headerCell = headerRow.createCell(4);
81          headerCell.setCellValue("commodityImpacted");
82
83          headerCell = headerRow.createCell(5);
84          headerCell.setCellValue("ministry");
85
86          headerCell = headerRow.createCell(6);
87          headerCell.setCellValue("port");
88
89          headerCell = headerRow.createCell(7);
90          headerCell.setCellValue("details");
91
92          headerCell = headerRow.createCell(8);
93          headerCell.setCellValue("sender_Name");
```

ExcelGeneratorClass.java

```
93          headerCell.setCellValue("sender_Name");
94
95          headerCell = headerRow.createCell(9);
96          headerCell.setCellValue("UPDATED_ON");
97
98          headerCell = headerRow.createCell(10);
99          headerCell.setCellValue("CREATED_ON");
100
101         headerCell = headerRow.createCell(11);
102         headerCell.setCellValue("USER_REF_NO");
103
104     }
105
106     private void writeDataLines(ResultSet result, XSSFWorkbook workbook,
107             XSSFSheet sheet) throws SQLException {
108         int rowCount = 1;
109
110         System.out.println("inside writeDataLines");
111
112         while (result.next()) {
113             String name = result.getString("name");
114             String iec = result.getString("iec");
115             String email = result.getString("email");
116             String mobile = result.getString("mobile");
117             String commodityImpacted = result.getString("commodityImpacted");
118             String ministry = result.getString("ministry");
119             String port = result.getString("port");
120             String details = result.getString("details");
121             String sender_Name = result.getString("sender_Name");
122             String UPDATED_ON = result.getString("UPDATED_ON");
123             String CREATED_ON = result.getString("CREATED_ON");
124             String USER_REF_NO = result.getString("USER_REF_NO");
125
126             Row row = sheet.createRow(rowCount++);
127
128             int columnCount = 0;
129             Cell cell = row.createCell(columnCount++);
130             cell.setCellValue(name);
131
132             cell = row.createCell(columnCount++);
133             cell.setCellValue(iec);
134
135             cell = row.createCell(columnCount++);
136             cell.setCellValue(email);
137
138             cell = row.createCell(columnCount++);
139             cell.setCellValue(mobile);
```