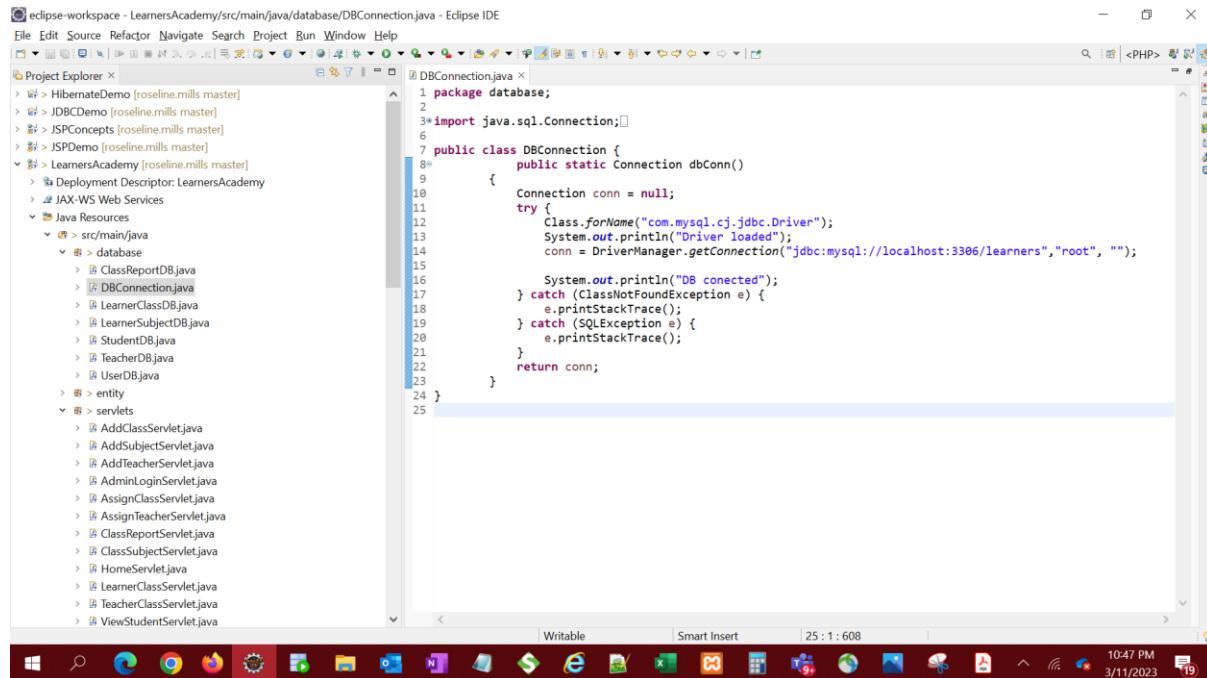


Learners Academy Source Code

Database

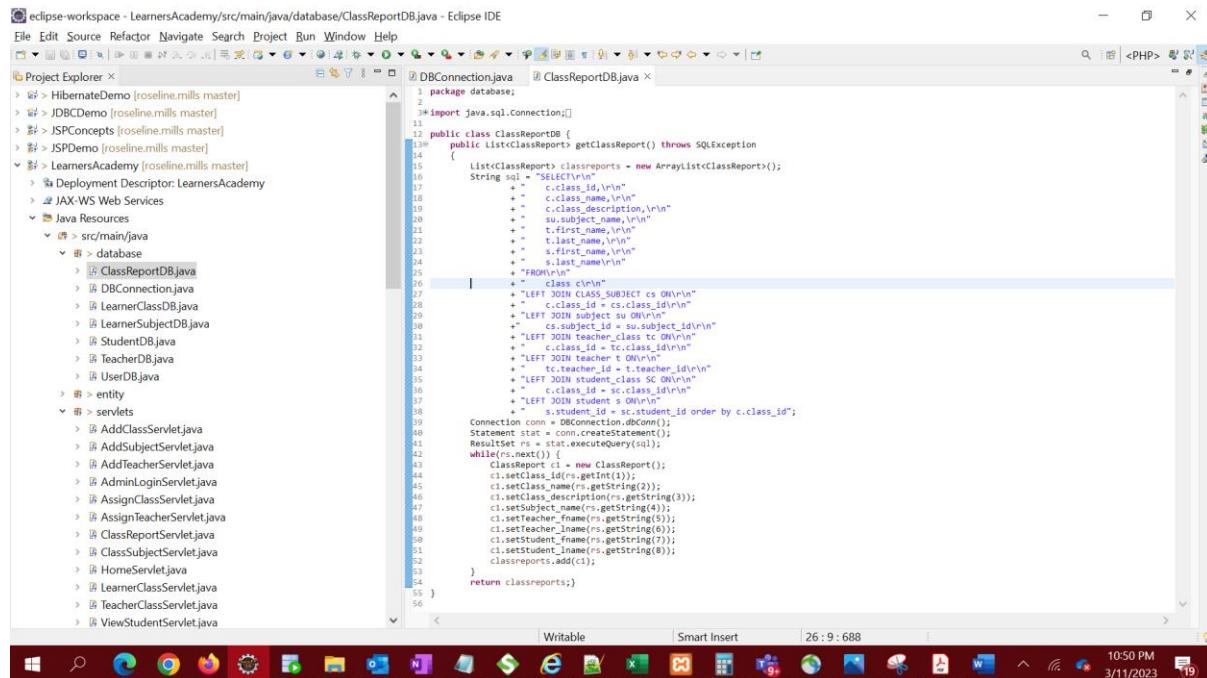
Database connection



The screenshot shows the Eclipse IDE interface with the DBConnection.java file open in the editor. The code defines a static Connection method named dbConn() that uses the MySQL JDBC driver to connect to a local host at port 3306. The code includes try-catch blocks for various SQLExceptions.

```
package database;
import java.sql.Connection;
public class DBConnection {
    public static Connection dbConn() {
        Connection conn = null;
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            System.out.println("Driver loaded");
            conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/learners","root", "");
            System.out.println("DB connected");
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return conn;
    }
}
```

ClassReportDB class



The screenshot shows the Eclipse IDE interface with the ClassReportDB.java file open in the editor. The code implements a getClassReport() method that constructs a SQL query to join four tables: CLASS, SUBJECT, TEACHER, and STUDENT. It uses LEFT JOIN clauses and ORDER BY clause to sort the results by class ID. The query selects columns for class ID, name, description, subject name, teacher first name, teacher last name, student first name, and student last name.

```
public class ClassReportDB {
    public List<ClassReport> getClassReport() throws SQLException {
        List<ClassReport> classeReports = new ArrayList<ClassReport>();
        String sql = "SELECT c.class_id, \r\n"
                    + "c.class_name, \r\n"
                    + "c.class_description, \r\n"
                    + "su.subject_name, \r\n"
                    + "tc.teacher_fname, \r\n"
                    + "tc.teacher_lname, \r\n"
                    + "s.first_name, \r\n"
                    + "s.last_name \r\n"
                    + "FROM \r\n"
                    + "class c \r\n"
                    + "LEFT JOIN CLASS_SUBJECT cs ON \r\n"
                    + "c.class_id = cs.class_id \r\n"
                    + "LEFT JOIN SUBJECT su ON \r\n"
                    + "cs.subject_id = su.subject_id \r\n"
                    + "LEFT JOIN teacher_class tc ON \r\n"
                    + "c.class_id = tc.class_id \r\n"
                    + "LEFT JOIN teacher t ON \r\n"
                    + "tc.teacher_id = t.teacher_id \r\n"
                    + "LEFT JOIN student_class SC ON \r\n"
                    + "c.class_id = sc.class_id \r\n"
                    + "LEFT JOIN student s ON \r\n"
                    + "sc.student_id = s.student_id order by c.class_id";
        Connection conn = DBConnection.dbConn();
        Statement stat = conn.createStatement();
        ResultSet rs = stat.executeQuery(sql);
        while(rs.next()){
            ClassReport cl = new ClassReport();
            cl.setClass_Id(rs.getInt(1));
            cl.setClass_Name(rs.getString(2));
            cl.setClass_Description(rs.getString(3));
            cl.setSubject_Name(rs.getString(4));
            cl.setTeacher_Fname(rs.getString(5));
            cl.setTeacher_Lname(rs.getString(6));
            cl.setStudent_FirstName(rs.getString(7));
            cl.setStudent_LastName(rs.getString(8));
            classeReports.add(cl);
        }
        return classeReports;
    }
}
```

LearnerClassDB Class

eclipse-workspace - LearnersAcademy/src/main/java/database/LearnerClassDB.java - Eclipse IDE

```

1 package database;
2
3 //import java.io.PrintWriter;
4 import java.sql.Connection;
5
6 public class LearnerClassDB {
7     public List<LearnerClass> getAllClasses() throws SQLException
8     {
9         List<LearnerClass> classes = new ArrayList<LearnerClass>();
10
11         String sql = "SELECT\r\n"
12             + "    c.class_id,\r\n"
13             + "    c.class_name,\r\n"
14             + "    c.class_description,\r\n"
15             + "    s.subject_name\r\n"
16             + "FROM\r\n"
17             + "    class c\r\n"
18             + "LEFT JOIN class_subject cs ON\r\n"
19             + "    cs.class_id = c.class_id\r\n"
20             + "LEFT JOIN SUBJECT s ON\r\n"
21             + "    s.subject_id = cs.subject_id";
22
23         Connection conn = DBConnection.dbConn();
24         Statement stat = conn.createStatement();
25         ResultSet rs = stat.executeQuery(sql);
26
27         while(rs.next()) { LearnerClass c1 = new LearnerClass();
28             c1.setClass_id(rs.getInt(1));
29             c1.setClass_name(rs.getString(2));
30             c1.setClass_description(rs.getString(3));
31             c1.setSubject_name(rs.getString(4));
32
33             classes.add(c1); }
34
35     }
36
37     return classes;
38 }
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
417
418
419
419
420
421
422
423
424
425
426
427
427
428
429
429
430
431
432
433
434
435
436
437
437
438
439
439
440
441
442
443
444
445
446
447
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
467
468
469
469
470
471
472
473
474
475
476
477
477
478
479
479
480
481
482
483
484
485
486
487
487
488
489
489
490
491
492
493
494
495
496
497
497
498
499
499
500
501
502
503
504
505
506
507
507
508
509
509
510
511
512
513
514
515
516
517
517
518
519
519
520
521
522
523
524
525
526
527
527
528
529
529
530
531
532
533
534
535
536
537
537
538
539
539
540
541
542
543
544
545
546
547
547
548
549
549
550
551
552
553
554
555
556
557
557
558
559
559
560
561
562
563
564
565
566
567
567
568
569
569
570
571
572
573
574
575
576
577
577
578
579
579
580
581
582
583
584
585
586
587
587
588
589
589
590
591
592
593
594
595
596
597
597
598
599
599
600
601
602
603
604
605
606
607
607
608
609
609
610
611
612
613
614
615
616
616
617
618
618
619
619
620
621
622
623
624
625
626
627
627
628
629
629
630
631
632
633
634
635
636
637
637
638
639
639
640
641
642
643
644
645
646
647
647
648
649
649
650
651
652
653
654
655
656
657
657
658
659
659
660
661
662
663
664
665
666
667
667
668
669
669
670
671
672
673
674
675
676
677
677
678
679
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
707
708
709
709
710
711
712
713
714
715
716
716
717
718
718
719
719
720
721
722
723
724
725
726
727
727
728
729
729
730
731
732
733
734
735
736
737
737
738
739
739
740
741
742
743
744
745
746
746
747
748
748
749
749
750
751
752
753
754
755
756
757
757
758
759
759
760
761
762
763
764
765
766
767
767
768
769
769
770
771
772
773
774
775
776
777
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
807
808
809
809
810
811
812
813
814
815
816
816
817
818
818
819
819
820
821
822
823
824
825
826
827
827
828
829
829
830
831
832
833
834
835
836
837
837
838
839
839
840
841
842
843
844
845
846
846
847
848
848
849
849
850
851
852
853
854
855
856
857
857
858
859
859
860
861
862
863
864
865
866
867
867
868
869
869
870
871
872
873
874
875
876
877
877
878
879
879
880
881
882
883
884
885
886
887
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
907
908
909
909
910
911
912
913
914
915
916
916
917
918
918
919
919
920
921
922
923
924
925
926
927
927
928
929
929
930
931
932
933
934
935
936
937
937
938
939
939
940
941
942
943
944
945
946
946
947
948
948
949
949
950
951
952
953
954
955
956
957
957
958
959
959
960
961
962
963
964
965
966
966
967
968
968
969
969
970
971
972
973
974
975
976
977
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
997
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1580
1581
1581
1582
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1590
1591
1591
1592
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1600
1601
1601
1602
1602
1603
1603
1604
1604
1605
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1610
1611
1611
1612
1612
1613
1613
1614
1614
1615
1615
1616
1616
1617
1617
1618
1618
1619
1619
1620
1620
1621
1621
1622
1622
1623
1623
1624
1624
1625
1625
1626
1626
1627
1627
1628
1628
1629
1629
1630
1630
1631
1631
1632
1632
1633
1633
1634
1634
1635
1635
1636
1636
1637
1637
1638
1638
1639
1639
1640
1640
164
```

LearnerSubjectDB Class

The screenshot shows the Eclipse IDE interface with the project 'LearnersAcademy' open. The 'Project Explorer' view on the left lists various Java files and resources. The 'LearnerSubjectDB.java' file is selected in the 'Editor' view on the right. The code implements a database access layer for learner subjects.

```
1 package database;
2
3 import java.sql.Connection;
4
5 public class LearnerSubjectDB {
6     public List<LearnerSubject> getAllSubjects() throws SQLException {
7         List<LearnerSubject> subjects = new ArrayList<LearnerSubject>();
8
9         String sql = "SELECT\r\n"
10            + "    s.subject_id,\r\n"
11            + "    s.subject_name,\r\n"
12            + "    c.class_name\r\n"
13            + "FROM SUBJECT\r\n"
14            + "    s\r\n"
15            + "LEFT JOIN class_subject cs ON\r\n"
16            + "    s.subject_id = cs.subject_id\r\n"
17            + "LEFT JOIN class c ON\r\n"
18            + "    cs.class_id = c.class_id";
19
20         Connection conn = DBConnection.dbConn();
21         Statement stat = conn.createStatement();
22         ResultSet rs = stat.executeQuery(sql);
23
24         while(rs.next()) {
25             LearnerSubject s2 = new LearnerSubject();
26             s2.setSubject_id(rs.getInt(1));
27             s2.setSubject_name(rs.getString(2));
28             s2.setClass_name(rs.getString(3));
29
30             subjects.add(s2);
31         }
32         return subjects;
33     }
34
35     public boolean AddSubjectUsingPreparedStatement(LearnerSubject s) throws SQLException {
36         String sql = "INSERT INTO subject VALUES (?,?)";
37
38         Connection conn = DBConnection.dbConn();
39         PreparedStatement stat = conn.prepareStatement(sql);
40         stat.setString(1, null);
41         stat.setString(2, s.getSubject_name());
42
43         try {
44             stat.executeUpdate();
45         } catch(Exception e) {
46             System.out.println("error "+e.getMessage());
47             return false;
48         }
49
50         return true;
51     }
52 }
```

The screenshot shows the continuation of the 'LearnerSubjectDB.java' code from the previous screenshot. The code handles the insertion of a new subject into the database using a prepared statement.

```
1 package database;
2
3 import java.sql.Connection;
4
5 public class LearnerSubjectDB {
6     public List<LearnerSubject> getAllSubjects() throws SQLException {
7         List<LearnerSubject> subjects = new ArrayList<LearnerSubject>();
8
9         String sql = "SELECT\r\n"
10            + "    s.subject_id,\r\n"
11            + "    s.subject_name,\r\n"
12            + "    c.class_name\r\n"
13            + "FROM SUBJECT\r\n"
14            + "    s\r\n"
15            + "LEFT JOIN class_subject cs ON\r\n"
16            + "    s.subject_id = cs.subject_id\r\n"
17            + "LEFT JOIN class c ON\r\n"
18            + "    cs.class_id = c.class_id";
19
20         Connection conn = DBConnection.dbConn();
21         Statement stat = conn.createStatement();
22         ResultSet rs = stat.executeQuery(sql);
23
24         while(rs.next()) {
25             LearnerSubject s2 = new LearnerSubject();
26             s2.setSubject_id(rs.getInt(1));
27             s2.setSubject_name(rs.getString(2));
28             s2.setClass_name(rs.getString(3));
29
30             subjects.add(s2);
31         }
32         return subjects;
33     }
34
35     public boolean AddSubjectUsingPreparedStatement(LearnerSubject s) throws SQLException {
36         String sql = "INSERT INTO subject VALUES (?,?)";
37
38         Connection conn = DBConnection.dbConn();
39         PreparedStatement stat = conn.prepareStatement(sql);
40         stat.setString(1, null);
41         stat.setString(2, s.getSubject_name());
42
43         try {
44             stat.executeUpdate();
45         } catch(Exception e) {
46             System.out.println("error "+e.getMessage());
47             return false;
48         }
49
50         return true;
51     }
52 }
```

StudentDB Class

The screenshot shows the Eclipse IDE interface with the StudentDB.java file open in the editor. The code implements a `StudentDB` class with a `getAllStudents()` method that performs a database query to fetch student information from multiple tables.

```
1 package database;
2 import java.sql.Connection;
3
4 public class StudentDB {
5     public List<Student> getAllStudents() throws SQLException
6     {
7         List<Student> students = new ArrayList<Student>();
8         String sql = "SELECT\r\n"
9             + " s.student_id,\r\n"
10            + " s.first_name,\r\n"
11            + " s.last_name,\r\n"
12            + " s.phone_number,\r\n"
13            + " s.email,\r\n"
14            + " c.class_name\r\n"
15            + " FROM\r\n"
16            + " student s\r\n"
17            + " LEFT JOIN student_class sc ON\r\n"
18            + " s.student_id = sc.student_id\r\n"
19            + " LEFT JOIN class c ON\r\n"
20            + " c.class_id = sc.class_id";
21
22         Connection conn = DBConnection.dbConn();
23         Statement stat = conn.createStatement();
24         ResultSet rs = stat.executeQuery(sql);
25         while(rs.next())
26         {
27             Student s1 = new Student();
28             s1.setStudent_id(rs.getInt(1));
29             s1.setFirst_name(rs.getString(2));
30             s1.setLast_name(rs.getString(3));
31             s1.setPhone_number(rs.getString(4));
32             s1.setEmail(rs.getString(5));
33             s1.setLearner_class(rs.getString(6));
34             students.add(s1);
35         }
36     }
37     return students;
38 }
39 }
```

TeacherDB Class

The screenshot shows the Eclipse IDE interface with the TeacherDB.java file open in the editor. The code implements a `TeacherDB` class with a `getallTeachers()` method that performs a database query to fetch teacher information from multiple tables.

```
1 package database;
2 import java.sql.Connection;
3 import java.sql.PreparedStatement;
4 import java.sql.ResultSet;
5 import java.sql.SQLException;
6 import java.util.List;
7 import java.util.ArrayList;
8 import entity.Teacher;
9
10 public class TeacherDB {
11     // fetch all records -> select *
12     public List<Teacher> getallTeachers() throws SQLException
13     {
14         List<Teacher> teachers = new ArrayList<Teacher>();
15         String sql = "SELECT\r\n"
16             + " t.teacher_id,\r\n"
17             + " t.first_name,\r\n"
18             + " t.last_name,\r\n"
19             + " t.phone_number,\r\n"
20             + " t.email,\r\n"
21             + " c.class_name\r\n"
22             + " FROM\r\n"
23             + " teacher t\r\n"
24             + " LEFT JOIN teacher_class tc ON\r\n"
25             + " t.teacher_id = tc.teacher_id\r\n"
26             + " LEFT JOIN class c ON\r\n"
27             + " tc.class_id = c.class_id";
28
29         Connection conn = DBConnection.dbConn();
30         Statement stat = conn.createStatement();
31         ResultSet rs = stat.executeQuery(sql);
32         while(rs.next())
33         {
34             Teacher t1 = new Teacher();
35             t1.setTeacher_id(rs.getInt(1));
36             t1.setFirst_name(rs.getString(2));
37             t1.setLast_name(rs.getString(3));
38             t1.setPhone_number(rs.getString(4));
39             t1.setEmail(rs.getString(5));
40             t1.setClass_name(rs.getString(6));
41             teachers.add(t1);
42         }
43     }
44     return teachers;
45 }
46 }
```

UserDB Class

The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - LearnersAcademy/src/main/java/database/UserDB.java - Eclipse IDE". The Project Explorer on the left lists various Java files and resources. The main editor window displays the following code for the UserDB class:

```
1 package database;
2
3 import java.sql.Connection;
4
5 public class UserDB {
6     public boolean validateUser(String username, String password) throws SQLException
7     {
8         boolean isValid = false;
9         String sql = "select password from user where username=?";
10        Connection conn = DBConnection.dbConn();
11        PreparedStatement stat = conn.prepareStatement(sql);
12        stat.setString(1, username);
13        ResultSet rs = stat.executeQuery();
14        if(rs.next()) {
15            String pass = rs.getString(1);
16            if(pass.equals(password))
17                isValid = true;
18        }
19    }
20    return isValid;
21 }
22 }
23 }
24 }
25 }
```

The status bar at the bottom shows "Writable" and "Smart Insert" mode, along with the current time and date.

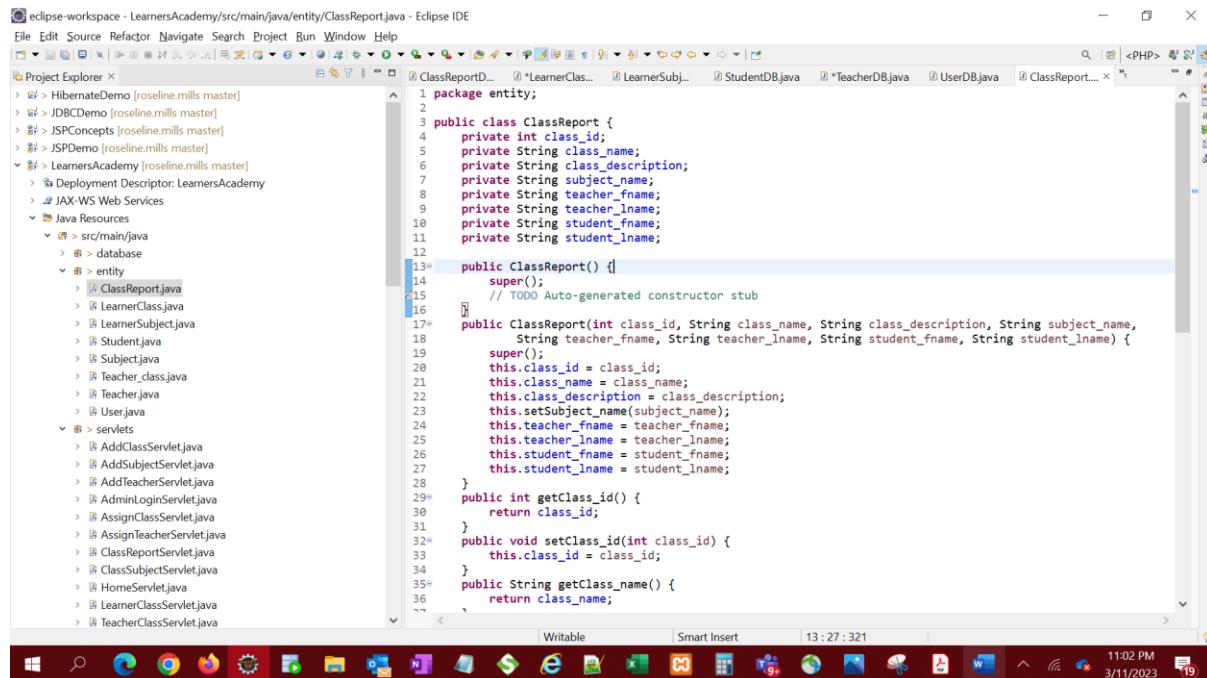
The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - LearnersAcademy/src/main/java/database/TeacherDB.java - Eclipse IDE". The Project Explorer on the left lists various Java files and resources. The main editor window displays the following code for the TeacherDB class:

```
47 public boolean AssignTeacherUsingPreparedStatement(String class_id, String teacher_id) throws SQLException
48 {
49     String sql = "INSERT INTO teacher_class VALUES (?, ?)";
50     System.out.println(sql);
51     // 1. DB connection
52     Connection conn = DBConnection.dbConn();
53     //2. create the statement
54     PreparedStatement stat = conn.prepareStatement(sql);
55     stat.setString(1, class_id);
56     stat.setString(2, teacher_id);
57     try {
58         stat.execute();
59     } catch (Exception e) {
60         System.out.println("error "+e.getMessage());
61         return false;
62     }
63     return true;
64 }
65
66
67 public boolean AddTeacherUsingPreparedStatement(Teacher teacher) throws SQLException
68 {
69     String sql = "INSERT INTO teacher VALUES (?, ?, ?, ?, ?)";
70     System.out.println(sql);
71     Connection conn = DBConnection.dbConn();
72     PreparedStatement stat = conn.prepareStatement(sql);
73     stat.setString(1, (null));
74     stat.setString(2, teacher.getFirst_name());
75     stat.setString(3, teacher.getLast_name());
76     stat.setString(4, teacher.getMember());
77     stat.setString(5, teacher.getEmail());
78     try {
79         stat.execute();
80     } catch (Exception e) {
81         System.out.println("error "+e.getMessage());
82         return false;
83     }
84
85
86
87
88
89 }
```

The status bar at the bottom shows "Writable" and "Smart Insert" mode, along with the current time and date.

Entity

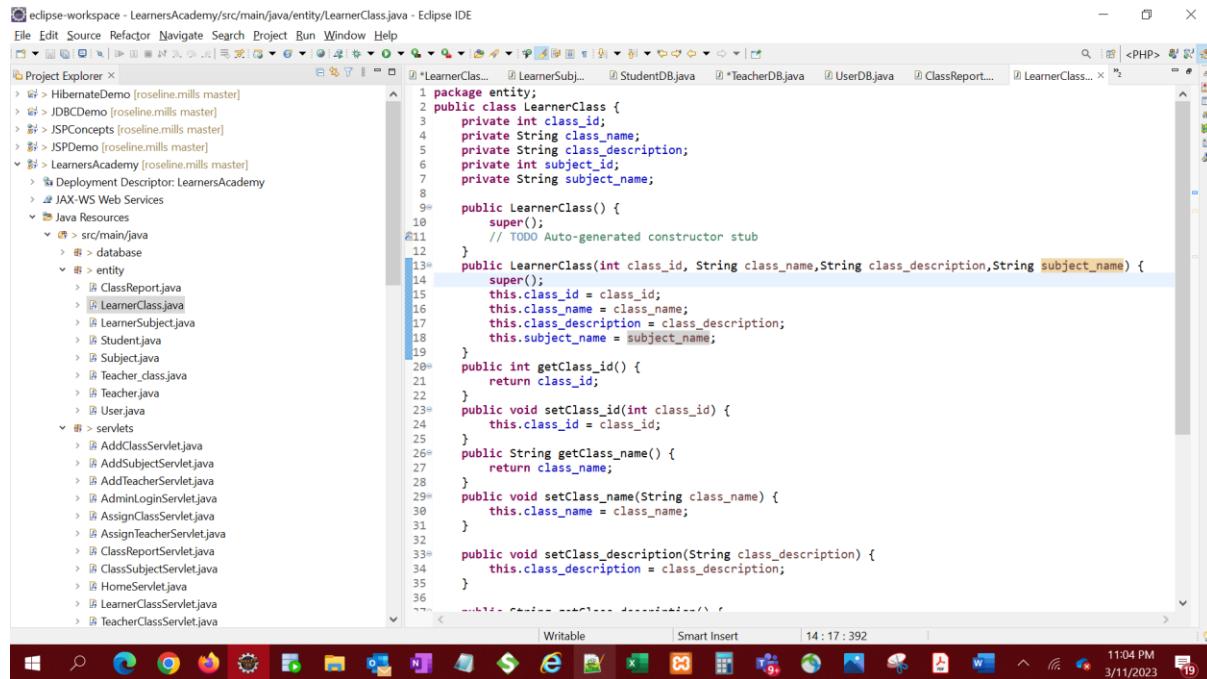
ClassReport Class



The screenshot shows the Eclipse IDE interface with the 'ClassReport.java' file open in the editor. The code defines a class 'ClassReport' with fields for class_id, class_name, class_description, subject_name, teacher_fname, teacher_lname, student_fname, and student_lname. It includes a constructor, getters, and setters for these fields.

```
1 package entity;
2
3 public class ClassReport {
4     private int class_id;
5     private String class_name;
6     private String class_description;
7     private String subject_name;
8     private String teacher_fname;
9     private String teacher_lname;
10    private String student_fname;
11    private String student_lname;
12
13    public ClassReport() {
14        super();
15        // TODO Auto-generated constructor stub
16    }
17    public ClassReport(int class_id, String class_name, String class_description, String subject_name,
18                      String teacher_fname, String teacher_lname, String student_fname, String student_lname) {
19        super();
20        this.class_id = class_id;
21        this.class_name = class_name;
22        this.class_description = class_description;
23        this.setSubject_name(subject_name);
24        this.teacher_fname = teacher_fname;
25        this.teacher_lname = teacher_lname;
26        this.student_fname = student_fname;
27        this.student_lname = student_lname;
28    }
29    public int getClass_id() {
30        return class_id;
31    }
32    public void setClass_id(int class_id) {
33        this.class_id = class_id;
34    }
35    public String getClass_name() {
36        return class_name;
37    }
38    public void setClass_name(String class_name) {
39        this.class_name = class_name;
40    }
41    public String getClass_description() {
42        return class_description;
43    }
44    public void setClass_description(String class_description) {
45        this.class_description = class_description;
46    }
47}
```

LearnerClass Class



The screenshot shows the Eclipse IDE interface with the 'LearnerClass.java' file open in the editor. The code defines a class 'LearnerClass' with fields for class_id, class_name, class_description, and subject_name. It includes a constructor, getters, and setters for these fields.

```
1 package entity;
2
3 public class LearnerClass {
4     private int class_id;
5     private String class_name;
6     private String class_description;
7     private int subject_id;
8     private String subject_name;
8
9     public LearnerClass() {
10        super();
11        // TODO Auto-generated constructor stub
12    }
13    public LearnerClass(int class_id, String class_name, String class_description, String subject_name) {
14        super();
15        this.class_id = class_id;
16        this.class_name = class_name;
17        this.class_description = class_description;
18        this.subject_name = subject_name;
19    }
20    public int getClass_id() {
21        return class_id;
22    }
23    public void setClass_id(int class_id) {
24        this.class_id = class_id;
25    }
26    public String getClass_name() {
27        return class_name;
28    }
29    public void setClass_name(String class_name) {
30        this.class_name = class_name;
31    }
32    public String getClass_description() {
33        return class_description;
34    }
35    public void setClass_description(String class_description) {
36        this.class_description = class_description;
37    }
38}
```

eclipse-workspace - LearnersAcademy/src/main/java/entity/LearnerClass.java - Eclipse IDE

```

1 package entity;
2
3 public class LearnerClass {
4     private int class_id;
5     private String class_name;
6     private String class_description;
7
8     public LearnerClass() {
9         super();
10    }
11
12     public int getClass_id() {
13         return class_id;
14     }
15
16     public void setClass_id(int class_id) {
17         this.class_id = class_id;
18     }
19
20     public String getClass_name() {
21         return class_name;
22     }
23
24     public void setClass_name(String class_name) {
25         this.class_name = class_name;
26     }
27
28     public String getClass_description() {
29         return class_description;
30     }
31
32     public void setClass_description(String class_description) {
33         this.class_description = class_description;
34     }
35
36     public String getSubject_id() {
37         return subject_id;
38     }
39
40     public int getSubject_id() {
41         return subject_id;
42     }
43
44     public void setSubject_id(int subject_id) {
45         this.subject_id = subject_id;
46     }
47
48     public void setSubject_name(String subject_name) {
49         this.subject_name = subject_name;
50     }
51
52     public String getSubject_name() {
53         return subject_name;
54     }
55 }

```

LearnerSubject.java Class

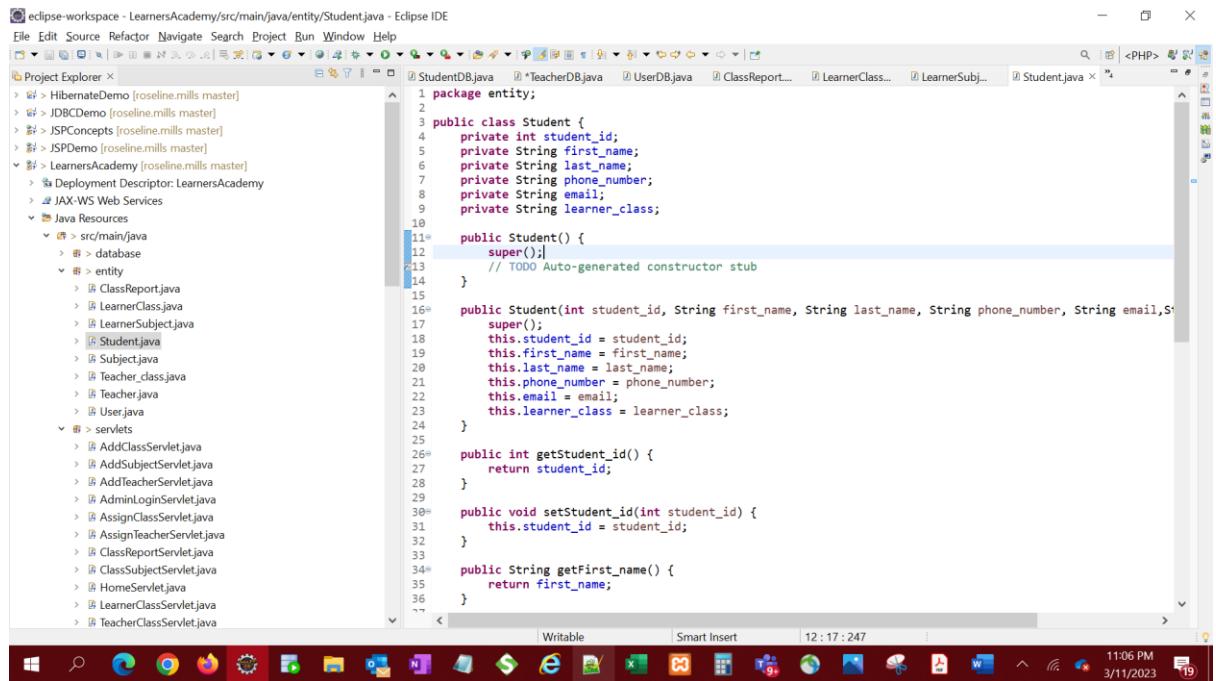
eclipse-workspace - LearnersAcademy/src/main/java/entity/LearnerSubject.java - Eclipse IDE

```

1 package entity;
2
3 public class LearnerSubject {
4     private int subject_id;
5     private String subject_name;
6     private String class_name;
7
8     public LearnerSubject(int subject_id, String subject_name, String class_name) {
9         super();
10    this.subject_id = subject_id;
11    this.subject_name = subject_name;
12    this.class_name = class_name;
13 }
14
15     public LearnerSubject() {
16         super();
17         // TODO Auto-generated constructor stub
18     }
19
20     public int getSubject_id() {
21         return subject_id;
22     }
23
24     public void setSubject_id(int subject_id) {
25         this.subject_id = subject_id;
26     }
27
28     public String getSubject_name() {
29         return subject_name;
30     }
31
32     public void setSubject_name(String subject_name) {
33         this.subject_name = subject_name;
34     }
35
36     public String getClass_name() {
37         return class_name;
38     }
39
40     public void setClass_name(String class_name) {
41         this.class_name = class_name;
42     }
43 }

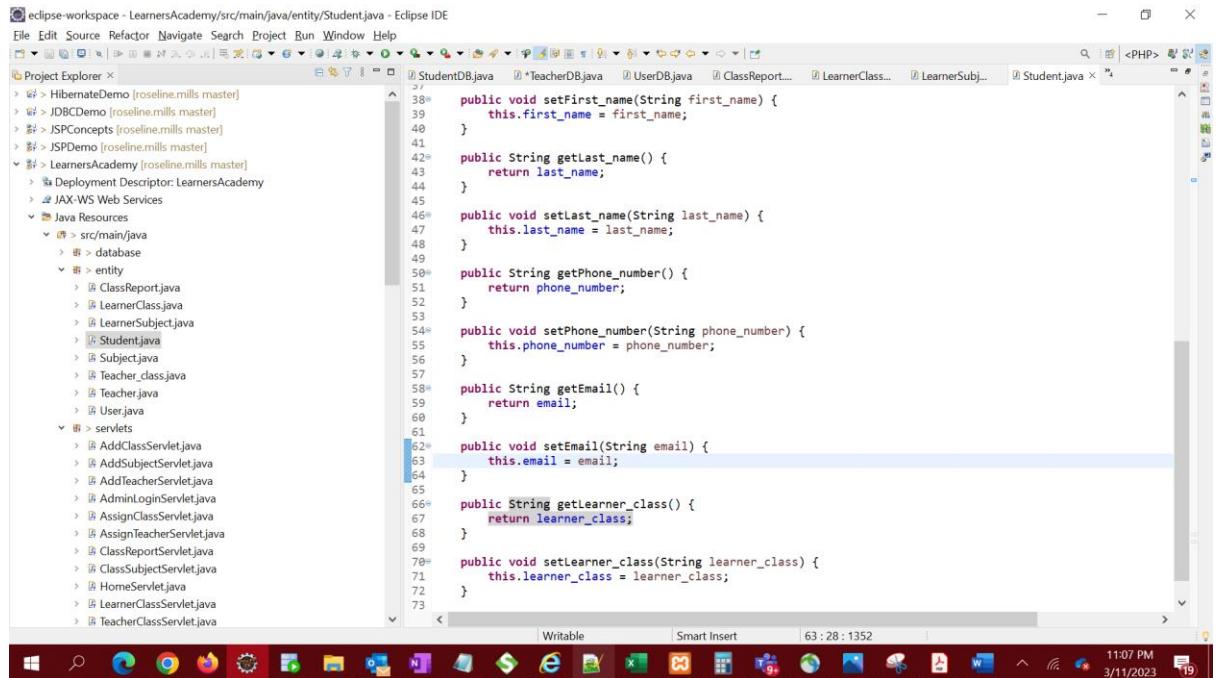
```

Student.java Class



The screenshot shows the Eclipse IDE interface with the Student.java file open in the editor. The code defines a Student entity with fields for student_id, first_name, last_name, phone_number, email, and learner_class. It includes constructor stubs and getter/setter methods.

```
1 package entity;
2
3 public class Student {
4     private int student_id;
5     private String first_name;
6     private String last_name;
7     private String phone_number;
8     private String email;
9     private String learner_class;
10
11    public Student() {
12        super();
13        // TODO Auto-generated constructor stub
14    }
15
16    public Student(int student_id, String first_name, String last_name, String phone_number, String email, String learner_class) {
17        super();
18        this.student_id = student_id;
19        this.first_name = first_name;
20        this.last_name = last_name;
21        this.phone_number = phone_number;
22        this.email = email;
23        this.learner_class = learner_class;
24    }
25
26    public int getStudent_id() {
27        return student_id;
28    }
29
30    public void setStudent_id(int student_id) {
31        this.student_id = student_id;
32    }
33
34    public String getFirst_name() {
35        return first_name;
36    }
37
38    public void setFirst_name(String first_name) {
39        this.first_name = first_name;
40    }
41
42    public String getLast_name() {
43        return last_name;
44    }
45
46    public void setLast_name(String last_name) {
47        this.last_name = last_name;
48    }
49
50    public String getPhone_number() {
51        return phone_number;
52    }
53
54    public void setPhone_number(String phone_number) {
55        this.phone_number = phone_number;
56    }
57
58    public String getEmail() {
59        return email;
60    }
61
62    public void setEmail(String email) {
63        this.email = email;
64    }
65
66    public String getLearner_class() {
67        return learner_class;
68    }
69
70    public void setLearner_class(String learner_class) {
71        this.learner_class = learner_class;
72    }
73}
```



The screenshot shows the Eclipse IDE interface with the Student.java file open in the editor, continuing from the previous code block. It includes additional getter and setter methods for the learner_class field.

```
1 package entity;
2
3 public class Student {
4     private int student_id;
5     private String first_name;
6     private String last_name;
7     private String phone_number;
8     private String email;
9     private String learner_class;
10
11    public Student() {
12        super();
13        // TODO Auto-generated constructor stub
14    }
15
16    public Student(int student_id, String first_name, String last_name, String phone_number, String email, String learner_class) {
17        super();
18        this.student_id = student_id;
19        this.first_name = first_name;
20        this.last_name = last_name;
21        this.phone_number = phone_number;
22        this.email = email;
23        this.learner_class = learner_class;
24    }
25
26    public int getStudent_id() {
27        return student_id;
28    }
29
30    public void setStudent_id(int student_id) {
31        this.student_id = student_id;
32    }
33
34    public String getFirst_name() {
35        return first_name;
36    }
37
38    public void setFirst_name(String first_name) {
39        this.first_name = first_name;
40    }
41
42    public String getLast_name() {
43        return last_name;
44    }
45
46    public void setLast_name(String last_name) {
47        this.last_name = last_name;
48    }
49
50    public String getPhone_number() {
51        return phone_number;
52    }
53
54    public void setPhone_number(String phone_number) {
55        this.phone_number = phone_number;
56    }
57
58    public String getEmail() {
59        return email;
60    }
61
62    public void setEmail(String email) {
63        this.email = email;
64    }
65
66    public String getLearner_class() {
67        return learner_class;
68    }
69
70    public void setLearner_class(String learner_class) {
71        this.learner_class = learner_class;
72    }
73}
```

User.java class

The screenshot shows the Eclipse IDE interface with the User.java file open in the editor. The code defines a User entity with private attributes for username and password, and methods for getting and setting these values. It also includes a constructor and an overridden toString() method.

```
1 package entity;
2 public class User {
3     private String username;
4     private String password;
5
6     public User() {}
7
8 }
9
10 public String getUsername() {
11     return username;
12 }
13
14 public void setUsername(String username) {
15     this.username = username;
16 }
17
18 public String getPassword() {
19     return password;
20 }
21
22 public void setPassword(String password) {
23     this.password = password;
24 }
25
26 public User(String username, String password) {
27     super();
28     this.username = username;
29     this.password = password;
30 }
31
32 @Override
33 public String toString() {
34     return "User [username=" + username + ", password=" + password + "]";
35 }
36 }
```

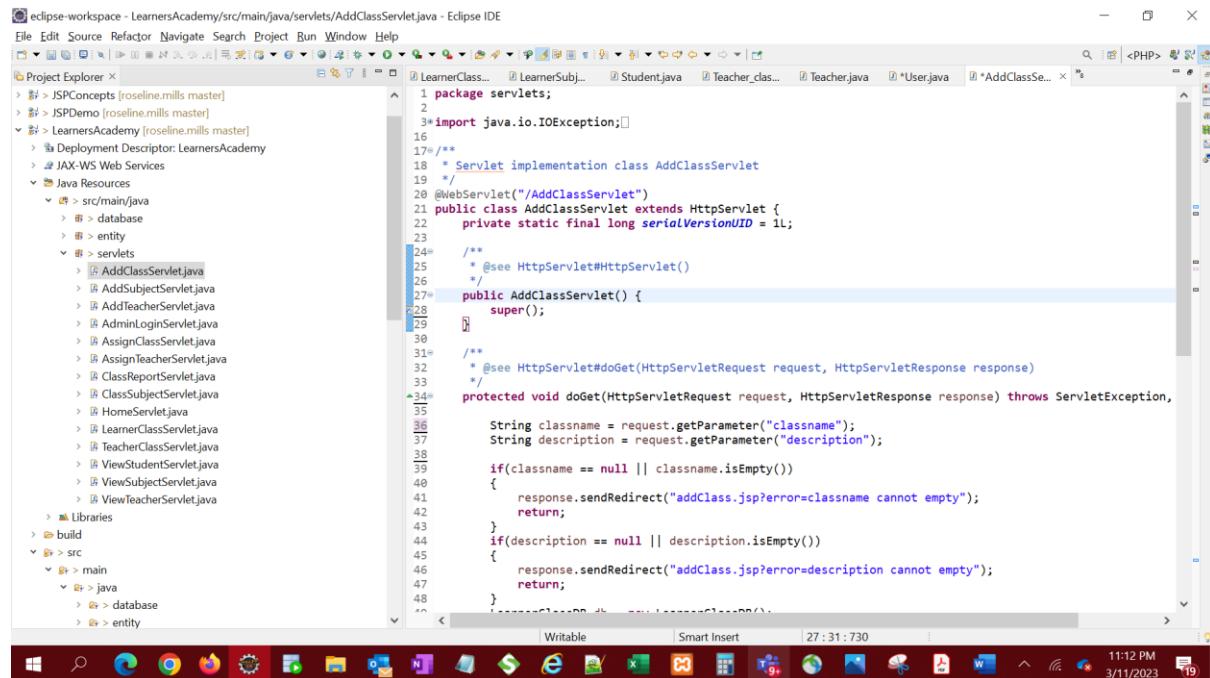
Teacher.java

The screenshot shows the Eclipse IDE interface with the Teacher.java file open in the editor. The code defines a Teacher entity with private attributes for teacher_id, first_name, last_name, phone_number, email, and class_name. It includes a constructor, getters, and setters for these attributes.

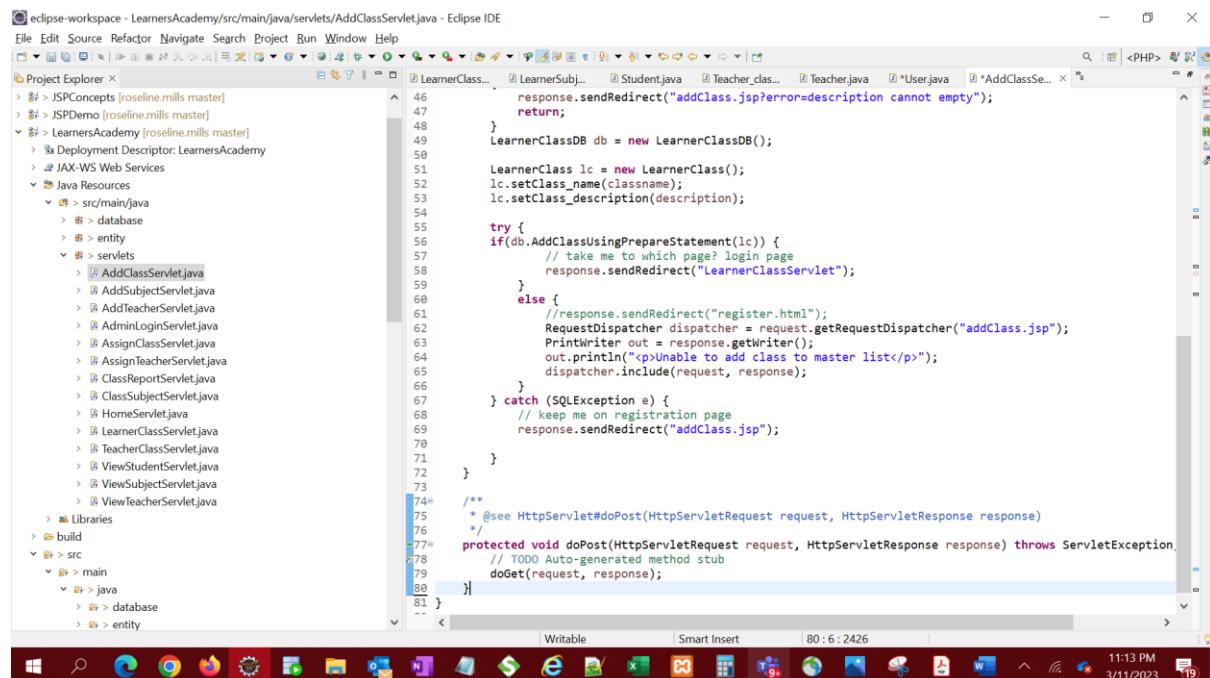
```
1 package entity;
2
3 public class Teacher {
4     private int teacher_id;
5     private String first_name;
6     private String last_name;
7     private String phone_number;
8     private String email;
9     private String class_name;
10
11     public Teacher() {
12         super();
13         // TODO Auto-generated constructor stub
14     }
15     public Teacher(int teacher_id, String first_name, String last_name, String phone_number, String email, String class_name) {
16         super();
17         this.teacher_id = teacher_id;
18         this.first_name = first_name;
19         this.last_name = last_name;
20         this.phone_number = phone_number;
21         this.email = email;
22         this.class_name = class_name;
23     }
24     public int getTeacher_id() {
25         return teacher_id;
26     }
27     public void setTeacher_id(int teacher_id) {
28         this.teacher_id = teacher_id;
29     }
30     public String getFirst_name() {
31         return first_name;
32     }
33     public void setFirst_name(String first_name) {
34         this.first_name = first_name;
35     }
36 }
```

Servlets

AddClassServlet.java Class



```
1 package servlets;
2
3 import java.io.IOException;
4
5 /**
6  * Servlet implementation class AddClassServlet
7  */
8 @WebServlet("/AddClassServlet")
9 public class AddClassServlet extends HttpServlet {
10     private static final long serialVersionUID = 1L;
11
12     /**
13      * @see HttpServlet#HttpServlet()
14     */
15     public AddClassServlet() {
16         super();
17     }
18
19     /**
20      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
21     */
22     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
23     {
24         /**
25          * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
26         */
27         String classname = request.getParameter("classname");
28         String description = request.getParameter("description");
29
30         if(classname == null || classname.isEmpty())
31         {
32             response.sendRedirect("addClass.jsp?error=classname cannot empty");
33             return;
34         }
35         if(description == null || description.isEmpty())
36         {
37             response.sendRedirect("addClass.jsp?error=description cannot empty");
38             return;
39         }
40     }
41 }
```



```
42
43     /**
44      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
45     */
46     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
47     {
48         /**
49          * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
50         */
51         LearnerClassDB db = new LearnerClassDB();
52
53         LearnerClass lc = new LearnerClass();
54         lc.setClass_name(classname);
55         lc.setClass_description(description);
56
57         try {
58             if(db.AddClassUsingPreparedStatement(lc)) {
59                 // take me to which page? login page
60                 response.sendRedirect("LearnerClassServlet");
61             }
62             else {
63                 //response.sendRedirect("register.html");
64                 RequestDispatcher dispatcher = request.getRequestDispatcher("addClass.jsp");
65                 PrintWriter out = response.getWriter();
66                 out.println("<p>Unable to add class to master list</p>");
67                 dispatcher.include(request, response);
68             }
69         } catch (SQLException e) {
70             // keep me on registration page
71             response.sendRedirect("addClass.jsp");
72         }
73     }
74
75     /**
76      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
77     */
78     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
79     {
80         /**
81         */
82     }
83 }
```

AddSubjectServlet.java Class

The screenshot shows the Eclipse IDE interface with the AddSubjectServlet.java file open in the editor. The code implements a HttpServlet for adding subjects. It includes imports for java.io.IOException and javax.servlet.ServletException, and extends HttpServlet. The doGet method checks for subjectname and sends a redirect if it's null or empty. It also initializes a LearnerSubjectDB instance.

```
1 package servlets;
2
3 import java.io.IOException;
4
5 /**
6  * Servlet implementation class AddSubjectServlet
7  */
8 @WebServlet("/AddSubjectServlet")
9 public class AddSubjectServlet extends HttpServlet {
10     private static final long serialVersionUID = 1L;
11
12     /**
13      * @see HttpServlet#HttpServlet()
14     */
15     public AddSubjectServlet() {
16         super();
17         // TODO Auto-generated constructor stub
18     }
19
20     /**
21      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
22     */
23     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
24         // TODO Auto-generated method stub
25         //response.getWriter().append("Served at: ").append(request.getContextPath());
26
27         String subjectname = request.getParameter("Subjectname");
28
29         if(subjectname == null || subjectname.isEmpty())
30             {
31                 response.sendRedirect("addSubject.jsp?error=subject cannot be empty");
32             }
33
34         LearnerSubjectDB db = new LearnerSubjectDB();
35
36     }
37
38 }
```

AddTeacherServlet.java class

The screenshot shows the Eclipse IDE interface with the AddTeacherServlet.java file open in the editor. The code implements a HttpServlet for adding teachers. It checks for firstname, lastname, email, and phononenumber parameters. If any are null or empty, it sets an error attribute and forwards the request to addTeacher.jsp. Otherwise, it initializes a RequestDispatcher and forwards the request to register.jsp.

```
1 package servlets;
2
3 import java.io.IOException;
4
5 /**
6  * Servlet implementation class AddTeacherServlet
7  */
8 @WebServlet("/AddTeacherServlet")
9 public class AddTeacherServlet extends HttpServlet {
10     private static final long serialVersionUID = 1L;
11
12     /**
13      * @see HttpServlet#HttpServlet()
14     */
15     public AddTeacherServlet() {
16         super();
17         // TODO Auto-generated constructor stub
18     }
19
20     /**
21      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
22     */
23     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
24
25         String firstname = request.getParameter("firstname");
26         String lastname = request.getParameter("lastname");
27         String email = request.getParameter("email");
28         String phononenumber = request.getParameter("phononenumber");
29
30         if(firstname == null || firstname.isEmpty())
31             {
32                 RequestDispatcher dispatcher = request.getRequestDispatcher("addTeacher.jsp");
33                 request.setAttribute("error", "firstname cannot be null");
34                 request.setAttribute("lastname", lastname);
35                 request.setAttribute("phononenumber", phononenumber);
36                 request.setAttribute("email", email);
37                 dispatcher.forward(request, response);
38                 //response.sendRedirect("register.jsp?error=email cannot be null");
39             }
40
41         if(lastname == null || lastname.isEmpty())
42             {
43                 response.sendRedirect("addTeacher.jsp?error=lastname cannot be empty");
44             }
45
46     }
47
48 }
```

AddTeacherServlet.java Class

```

eclipse-workspace - LearnersAcademy/src/main/java/servlets/AddTeacherServlet.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer < PHP > AddTeacherServlet.java
> JSPConcepts [roseline.mills master]
> JPDemo [roseline.mills master]
> LearnersAcademy [roseline.mills master]
  Deployment Descriptor: LearnersAcademy
  JAX-WS Web Services
  Java Resources
    src/main/java
      database
      entity
      servlets
        AddClassServlet.java
        AddSubjectServlet.java
        AddTeacherServlet.java
        AdminLoginServlet.java
        AssignClassServlet.java
        AssignTeacherServlet.java
        ClassReportServlet.java
        ClassSubjectServlet.java
        HomeServlet.java
        LearnerClassServlet.java
        TeacherClassServlet.java
        ViewStudentServlet.java
        ViewSubjectServlet.java
        ViewTeacherServlet.java
  Libraries
  build
  src
    main
      java
        database
        entity
  Writable Smart Insert 85 : 17 : 2833
  11:15 PM 3/11/2023 19

  package servlets;
  import java.io.IOException;
  ...
  * Servlet implementation class AdminLoginServlet
  */
  @WebServlet("/AdminLoginServlet")
  public class AdminLoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    ...
    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
      // TODO Auto-generated method stub
      doGet(request, response);
    }
  }

  /**
   * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
   */
  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    String email = request.getParameter("email");
    String password = request.getParameter("password");
    response.setContentType("text/html");
    UserDB db = new UserDB();
    try {
      if(db.validateUser(email, password)) {
        request.setAttribute("username", email);
        HttpSession session = request.getSession();
        session.setAttribute("id", email);
        response.sendRedirect("dashboard.jsp");
      } else {
        response.sendRedirect("index.jsp?error=Invalid Credentials");
        RequestDispatcher dispatcher = request.getRequestDispatcher("index.jsp");
        PrintWriter out = response.getWriter();
        out.println("Invalid credentials");
        dispatcher.include(request, response);
      }
    } catch (Exception e) {
      e.printStackTrace();
      response.sendRedirect("index.jsp?error=something went wrong");
    }
  }
}

```

AdminLoginServlet.java Class

AdminLoginServlet.java Class

```

eclipse-workspace - LearnersAcademy/src/main/java/servlets/AdminLoginServlet.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer < PHP > AdminLoginServlet.java
> JSPConcepts [roseline.mills master]
> JPDemo [roseline.mills master]
> LearnersAcademy [roseline.mills master]
  Deployment Descriptor: LearnersAcademy
  JAX-WS Web Services
  Java Resources
    src/main/java
      database
      entity
      servlets
        AddClassServlet.java
        AddSubjectServlet.java
        AddTeacherServlet.java
        AdminLoginServlet.java
        AssignClassServlet.java
        AssignTeacherServlet.java
        ClassReportServlet.java
        ClassSubjectServlet.java
        HomeServlet.java
        LearnerClassServlet.java
        TeacherClassServlet.java
        ViewStudentServlet.java
        ViewSubjectServlet.java
        ViewTeacherServlet.java
  Libraries
  build
  src
    main
      java
        database
        entity
      servlets
        AddClassServlet.java
        AddSubjectServlet.java
        AddTeacherServlet.java
        AdminLoginServlet.java
        AssignClassServlet.java
        AssignTeacherServlet.java
        ClassReportServlet.java
        ClassSubjectServlet.java
        HomeServlet.java
  Writable Smart Insert 36 : 59 : 1096
  11:16 PM 3/11/2023 19

  package servlets;
  import java.io.IOException;
  ...
  * Servlet implementation class AdminLoginServlet
  */
  @WebServlet("/AdminLoginServlet")
  public class AdminLoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    ...
    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
      // TODO Auto-generated method stub
      doPost(request, response);
    }
  }

  /**
   * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
   */
  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    String username = request.getParameter("username");
    String password = request.getParameter("password");
    response.setContentType("text/html");
    UserDB db = new UserDB();
    try {
      if(db.validateUser(username, password)) {
        request.setAttribute("username", username);
        HttpSession session = request.getSession();
        session.setAttribute("id", username);
        response.sendRedirect("dashboard.jsp");
      } else {
        response.sendRedirect("index.jsp?error=Invalid Credentials");
        RequestDispatcher dispatcher = request.getRequestDispatcher("index.jsp");
        PrintWriter out = response.getWriter();
        out.println("Invalid credentials");
        dispatcher.include(request, response);
      }
    } catch (Exception e) {
      e.printStackTrace();
      response.sendRedirect("index.jsp?error=something went wrong");
    }
  }
}

```

AssignClassServlet.java Class

The screenshot shows the Eclipse IDE interface with the project 'eclipse-workspace - LearnersAcademy' open. The 'AssignClassServlet.java' file is selected in the Project Explorer. The code implements a HttpServlet for assigning classes. It uses DAOs for LearnerClassDB and LearnerSubjectDB to get lists of classes and subjects respectively. It then creates a RequestDispatcher for 'assignClass.jsp' and sets attributes for 'classes' and 'subjects'. Finally, it forwards the request to the dispatcher. The code includes comments and imports for java.io.IOException and javax.servlet.ServletException.

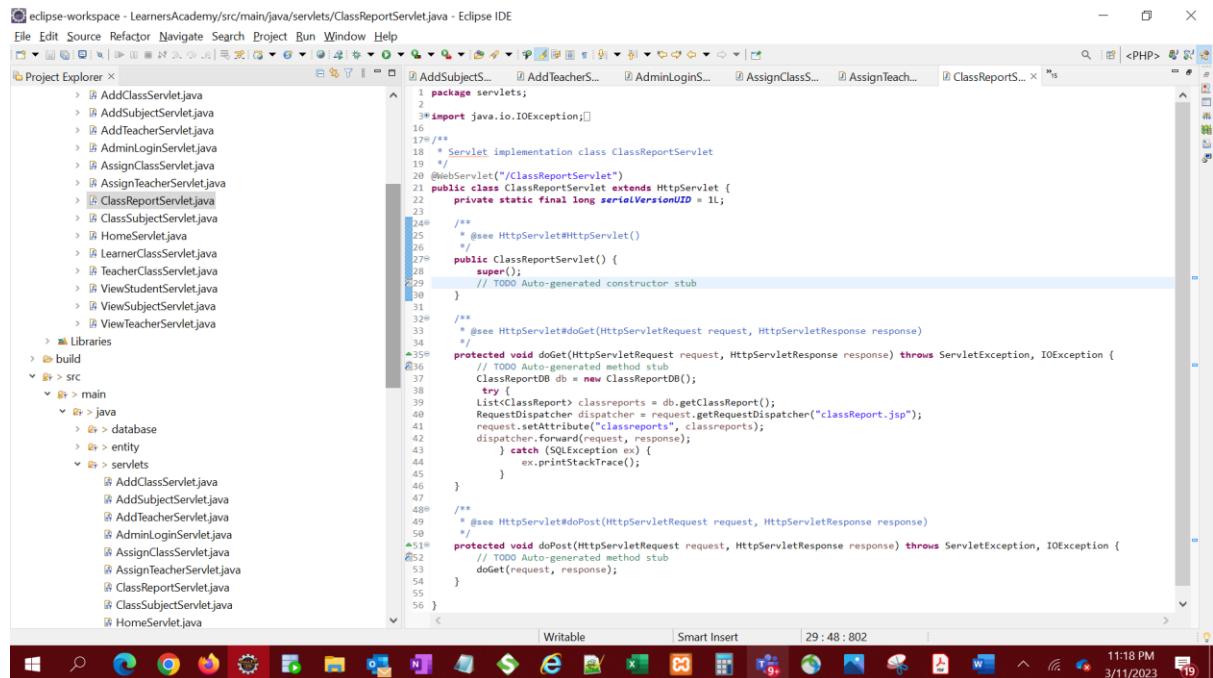
```
1 package servlets;
2
3 import java.io.IOException;
4
5 /**
6  * @see HttpServlet#HttpServlet()
7  */
8 public AssignClassServlet() {
9     super();
10    // TODO Auto-generated constructor stub
11 }
12
13 /**
14  * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
15 */
16 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
17     // TODO Auto-generated method stub
18
19     LearnerClassDB db = new LearnerClassDB();
20     try {
21         List<LearnerClass> classes = db.getAllClasses();
22         RequestDispatcher dispatcher = request.getRequestDispatcher("assignClass.jsp");
23         request.setAttribute("classes", classes);
24         dispatcher.include(request, response);
25     } catch (SQLException ex) {
26         ex.printStackTrace();
27     }
28 }
29
30 /**
31  * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
32 */
33 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
34     // TODO Auto-generated method stub
35     doGet(request, response);
36 }
```

AssignTeacherServlet.java Class

The screenshot shows the Eclipse IDE interface with the project 'eclipse-workspace - LearnersAcademy' open. The 'AssignTeacherServlet.java' file is selected in the Project Explorer. The code implements a HttpServlet for assigning teachers. It uses DAOs for TeacherDB and LearnerClassDB to get lists of teachers and learner classes respectively. It then creates a RequestDispatcher for 'assignTeacher.jsp' and sets attributes for 'teachers' and 'learnerClasses'. Finally, it forwards the request to the dispatcher. The code includes comments and imports for java.io.IOException and javax.servlet.ServletException.

```
1 package servlets;
2
3 import java.io.IOException;
4
5 /**
6  * @see HttpServlet#HttpServlet()
7  */
8 public AssignTeacherServlet() {
9     super();
10    // TODO Auto-generated constructor stub
11 }
12
13 /**
14  * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
15 */
16 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
17     // TODO Auto-generated method stub
18
19     LearnerClassDB db = new LearnerClassDB();
20     try {
21         List<LearnerClass> learnerClasses = db.getAllClasses();
22         RequestDispatcher dispatcher = request.getRequestDispatcher("assignTeacher.jsp");
23         request.setAttribute("learnerClasses", learnerClasses);
24         dispatcher.forward(request, response);
25     } catch (SQLException ex) {
26         ex.printStackTrace();
27     }
28 }
29
30 /**
31  * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
32 */
33 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
34     // TODO Auto-generated method stub
35     doGet(request, response);
36 }
```

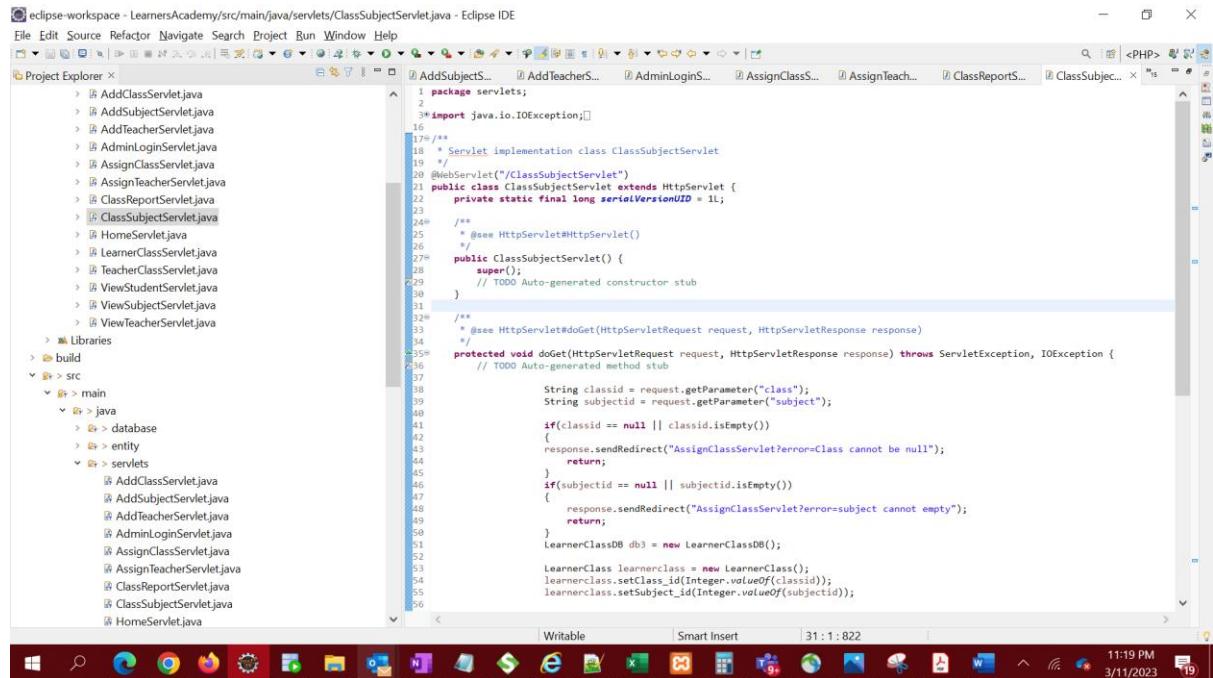
ClassReportServlet.java Class



The screenshot shows the Eclipse IDE interface with the project 'LearnersAcademy' open. The 'Project Explorer' view on the left lists various Java files under the 'src/main/java/servlets' package. The 'ClassReportServlet.java' file is selected and displayed in the central editor window. The code implements a servlet for generating class reports. It includes imports for java.io.IOException and javax.servlet.ServletException, and annotations for @WebServlet and @Override. The doGet() method retrieves class reports from a database and forwards them to a JSP page. The doPost() method is also implemented.

```
1 package servlets;
2
3 import java.io.IOException;
4
5 /**
6 * Servlet implementation class ClassReportServlet
7 */
8 @WebServlet("/ClassReportServlet")
9 public class ClassReportServlet extends HttpServlet {
10     private static final long serialVersionUID = 1L;
11
12     /**
13      * @see HttpServlet#HttpServlet()
14     */
15     public ClassReportServlet() {
16         super();
17         // TODO Auto-generated constructor stub
18     }
19
20     /**
21      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
22     */
23     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
24         // TODO Auto-generated method stub
25         ClassReportDB db = new ClassReportDB();
26         try {
27             List<ClassReport> classReports = db.getClassReport();
28             RequestDispatcher dispatcher = request.getRequestDispatcher("classReport.jsp");
29             request.setAttribute("classReports", classReports);
30             dispatcher.forward(request, response);
31         } catch (SQLException ex) {
32             ex.printStackTrace();
33         }
34     }
35
36     /**
37      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
38     */
39     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
40         // TODO Auto-generated method stub
41         doGet(request, response);
42     }
43 }
```

ClassSubjectServlet.java Class



The screenshot shows the Eclipse IDE interface with the project 'LearnersAcademy' open. The 'Project Explorer' view on the left lists various Java files under the 'src/main/java/servlets' package. The 'ClassSubjectServlet.java' file is selected and displayed in the central editor window. The code implements a servlet for managing class-subject assignments. It includes imports for java.io.IOException and javax.servlet.ServletException, and annotations for @WebServlet and @Override. The doGet() method handles requests to assign subjects to a specific class. It checks if both class and subject IDs are provided and then updates the LearnerClassDB with the new assignment.

```
1 package servlets;
2
3 import java.io.IOException;
4
5 /**
6 * Servlet implementation class ClassSubjectServlet
7 */
8 @WebServlet("/ClassSubjectServlet")
9 public class ClassSubjectServlet extends HttpServlet {
10     private static final long serialVersionUID = 1L;
11
12     /**
13      * @see HttpServlet#HttpServlet()
14     */
15     public ClassSubjectServlet() {
16         super();
17         // TODO Auto-generated constructor stub
18     }
19
20     /**
21      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
22     */
23     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
24         // TODO Auto-generated method stub
25         String classid = request.getParameter("class");
26         String subjectid = request.getParameter("subject");
27
28         if(classid == null || classid.isEmpty())
29         {
30             response.sendRedirect("AssignClassServlet?error=Class cannot be null");
31             return;
32         }
33         if(subjectid == null || subjectid.isEmpty())
34         {
35             response.sendRedirect("AssignClassServlet?error=subject cannot be empty");
36             return;
37         }
38         LearnerClassDB db3 = new LearnerClassDB();
39
40         LearnerClass learnerClass = new LearnerClass();
41         learnerClass.setClass_id(Integer.valueOf(classid));
42         learnerClass.setSubject_id(Integer.valueOf(subjectid));
43     }
44 }
```

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse-workspace - LearnersAcademy/src/main/java/servlets/ClassSubjectServlet.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar with icons for file operations.
- Project Explorer View (Left):**
 - Shows the project structure:
 - src
 - main
 - servlets
 - AddClassServlet.java
 - AddSubjectServlet.java
 - AddTeacherServlet.java
 - AdminLoginServlet.java
 - AssignClassServlet.java
 - AssignTeacherServlet.java
 - ClassReportServlet.java
 - ClassSubjectServlet.java
 - HomeServlet.java
 - Libraries
 - build
 - Code Editor View (Right):** Displays the Java code for ClassSubjectServlet.java. The code handles various HTTP requests for managing classes and subjects. It includes logic for adding subjects, teachers, and logging in administrators. It also handles errors such as null subject or class IDs and invalid registration attempts. A TODO comment is present for the doPost method.
 - Bottom Status Bar:** Shows Writable, Smart Insert, and the current timestamp (85 : 2 : 2720).
 - Taskbar:** Shows icons for various applications including Windows File Explorer, Task View, and several browser windows.
 - System Tray:** Shows the date (3/11/2023) and time (11:20 PM).

HomeServlet.java Class

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse-workspace - LearnersAcademy/src/main/java/servlets/HomeServlet.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar with icons for file operations.
- Project Explorer:** Shows the project structure:
 - src folder contains:
 - main folder
 - servlets folder containing AddClassServlet.java, AddSubjectServlet.java, AddTeacherServlet.java, AdminLoginServlet.java, AssignClassServlet.java, AssignTeacherServlet.java, ClassReportServlet.java, ClassSubjectServlet.java, and HomeServlet.java.
- Code Editor:** Displays the Java code for HomeServlet.java. The code includes imports for java.io.IOException, annotations for HttpServlet and @WebServlet("index"), and methods doGet and doPost. The code is annotated with TODO comments for stubs.
- Bottom Status Bar:** Shows Writable, Smart Insert, and the current time (46:6:1492).
- Taskbar:** Shows icons for various applications including Windows Start, File Explorer, Task View, and several browser and productivity icons.
- System Tray:** Shows the date (3/11/2023) and time (11:20 PM).

LearnerClassServlet.java Class

The screenshot shows the Eclipse IDE interface with the project 'eclipse-workspace - LearnersAcademy/src/main/java/servlets/HomeServlet.java' selected. The code editor displays the 'HomeServlet.java' file, which contains Java code for a HttpServlet. The code includes imports for java.io.IOException, annotations like @WebServlet("/Index"), and methods like doGet() and doPost(). The code is well-formatted with proper indentation and comments. The status bar at the bottom shows the time as 11:20 PM and the date as 3/11/2023.

```
1 package servlets;
2
3 import java.io.IOException;
4
5 /**
6  * Servlet implementation class index
7  */
8 @WebServlet("/Index")
9 public class HomeServlet extends HttpServlet {
10     private static final long serialVersionUID = 1L;
11
12     /**
13      * @see HttpServlet#HttpServlet()
14     */
15     public HomeServlet() {
16         super();
17         // TODO Auto-generated constructor stub
18     }
19
20     /**
21      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
22     */
23     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
24         // TODO Auto-generated method stub
25         response.getWriter().append("Served at: ").append(request.getContextPath());
26     }
27
28     /**
29      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
30     */
31     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
32         // TODO Auto-generated method stub
33         response.getWriter().append("Served at: ").append(request.getContextPath());
34     }
35
36     /**
37      * @see HttpServlet#doPut(HttpServletRequest request, HttpServletResponse response)
38     */
39     protected void doPut(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
40         // TODO Auto-generated method stub
41         String username = request.getParameter("username");
42         String password = request.getParameter("password");
43         PrintWriter out = response.getWriter();
44         response.setContentType("text/html");
45         out.println(username);
46         out.println(password);
47     }
48 }
```

TeacherClassServlet.java Class

The screenshot shows the Eclipse IDE interface with the project 'eclipse-workspace - LearnersAcademy/src/main/java/servlets/TeacherClassServlet.java' selected. The code editor displays the 'TeacherClassServlet.java' file, which contains Java code for a HttpServlet. The code includes imports for java.io.IOException, annotations like @WebServlet("/TeacherClassServlet"), and methods like doGet() and doPost(). The code is well-formatted with proper indentation and comments. The status bar at the bottom shows the time as 11:22 PM and the date as 3/11/2023.

```
1 package servlets;
2
3 import java.io.IOException;
4
5 /**
6  * Servlet implementation class Teacher_Class
7  */
8 @WebServlet("/TeacherClassServlet")
9 public class TeacherClassServlet extends HttpServlet {
10     private static final long serialVersionUID = 1L;
11
12     /**
13      * @see HttpServlet#HttpServlet()
14     */
15     public TeacherClassServlet() {
16         super();
17         // TODO Auto-generated constructor stub
18     }
19
20     /**
21      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
22     */
23     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
24         String teacherid = request.getParameter("teacher");
25         String classid = request.getParameter("class");
26
27         if(classid == null || classid.isEmpty()) {
28             response.sendRedirect("AssignTeacherServlet?error= Class cannot be null");
29             return;
30         }
31         if(teacherid == null || teacherid.isEmpty())
32             response.sendRedirect("AssignTeacherServlet?error= Teacher cannot empty");
33         return;
34     }
35
36     TeacherDB database = new TeacherDB();
37     Teacher teacher = new Teacher();
38     try {
39         if (database.AssignTeacherUsingPreparedStatement(teacherid,classid)) {
40             response.sendRedirect("ViewTeacherServlet");
41         } else {
42             RequestDispatcher dispatcher = request.getRequestDispatcher("AssignTeacherServlet");
43             PrintWriter out = response.getWriter();
44             out.println("<p>unable to assign teacher to class</p>");
45             dispatcher.include(out, response);
46         }
47     } catch (SQLException e) {
48         // TODO Auto-generated catch block
49         e.printStackTrace();
50     }
51
52     /**
53      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
54     */
55     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
56 }
```

ViewStudentServlet.java Class

eclipse-workspace - LearnersAcademy/src/main/java/servlets/ViewStudentServlet.java - Eclipse IDE

```

1 package servlets;
2 import java.io.IOException;
3 /**
4  * Servlet implementation class StudentServlet
5 */
6 @WebServlet("/studentServlet")
7 public class ViewStudentServlet extends HttpServlet {
8     private static final long serialVersionUID = 1L;
9 
10    /**
11     * @see HttpServlet#HttpServlet()
12     */
13    public ViewStudentServlet() {
14        super();
15        // TODO Auto-generated constructor stub
16    }
17 
18    /**
19     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
20     */
21    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
22        StudentDB db = new StudentDB();
23        try {
24            List<Student> students = db.getAllStudents();
25            RequestDispatcher dispatcher = request.getRequestDispatcher("viewStudents.jsp");
26            request.setAttribute("students", students);
27            dispatcher.forward(request, response);
28        } catch (SQLException ex) {
29            ex.printStackTrace();
30        }
31    }
32 
33    /**
34     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
35     */
36    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
37        StudentDB db = new StudentDB();
38        try {
39            List<Student> students = db.getAllStudents();
40            RequestDispatcher dispatcher = request.getRequestDispatcher("viewStudents.jsp");
41            request.setAttribute("students", students);
42            dispatcher.forward(request, response);
43        } catch (SQLException ex) {
44            ex.printStackTrace();
45        }
46        /**
47         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
48         */
49        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
50            // TODO Auto-generated method stub
51            doGet(request, response);
52        }
53    }
54 }
55

```

ViewSubjectServlet.java Class

eclipse-workspace - LearnersAcademy/src/main/java/servlets/ViewSubjectServlet.java - Eclipse IDE

```

1 package servlets;
2 import java.io.IOException;
3 /**
4  * Servlet implementation class SubjectServlet
5 */
6 @WebServlet("/subjectServlet")
7 public class ViewSubjectServlet extends HttpServlet {
8     private static final long serialVersionUID = 1L;
9 
10    /**
11     * @see HttpServlet#HttpServlet()
12     */
13    public ViewSubjectServlet() {
14        super();
15        // TODO Auto-generated constructor stub
16    }
17 
18    /**
19     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
20     */
21    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
22        LearnerSubjectDB db = new LearnerSubjectDB();
23        try {
24            List<LearnerSubject> subjects = db.getAllSubjects();
25            RequestDispatcher dispatcher = request.getRequestDispatcher("subjects.jsp");
26            request.setAttribute("subjects", subjects);
27            dispatcher.forward(request, response);
28        } catch (SQLException ex) {
29            ex.printStackTrace();
30        }
31    }
32 
33    /**
34     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
35     */
36    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
37        LearnerSubjectDB db = new LearnerSubjectDB();
38        try {
39            List<LearnerSubject> subjects = db.getAllSubjects();
40            RequestDispatcher dispatcher = request.getRequestDispatcher("subjects.jsp");
41            request.setAttribute("subjects", subjects);
42            dispatcher.forward(request, response);
43        } catch (SQLException ex) {
44            ex.printStackTrace();
45        }
46        /**
47         * ArrayList subjects1 = new ArrayList(); subjects1.get Object subjects =
48         * subjects1.get(0); //subjects1 = db.getAllSubjects();
49         * request.setAttribute("subjects", subjects);
50         */
51        //forward the request to jsp
52        //request.getRequestDispatcher("addSubject.jsp").forward(request, response);
53        /**
54         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
55         */
56        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
57            // TODO Auto-generated method stub
58            doGet(request, response);
59        }
60    }
61 }
62

```

ViewteacherServlet.java Class

eclipse-workspace - LearnersAcademy/src/main/java/servlets/ViewTeacherServlet.java - Eclipse IDE

```

1 package servlets;
2
3 import java.io.IOException;
4
5 /**
6 * Servlet implementation class teacherServlet
7 */
8 @WebServlet("/ViewTeacherServlet")
9 public class ViewTeacherServlet extends HttpServlet {
10     private static final long serialVersionUID = 1L;
11
12     /**
13      * @see HttpServlet#HttpServlet()
14     */
15     public ViewTeacherServlet() {
16         super();
17         // TODO Auto-generated constructor stub
18     }
19
20     /**
21      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
22     */
23     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
24         // TODO Auto-generated method stub
25         response.getWriter().append("Served at: ").append(request.getContextPath());
26     }
27
28     /**
29      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
30     */
31     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
32         // TODO Auto-generated method stub
33     }
34 }
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56 }
```

JSP

addClass.jsp

eclipse-workspace - LearnersAcademy/src/main/webapp/addClass.jsp - Eclipse IDE

```

1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2 pageEncoding="ISO-8859-1"%>
3 <%@include file="include/navigation.jsp"%>
4 <!DOCTYPE html>
5 <html>
6 <head>
7 <meta charset="ISO-8859-1">
8 <title>Add Class</title>
9 </head>
10 <body>
11 <div class="container" style="margin-top: 20px; margin-left: 50px;">
12     <h2 class="header">Add a class to master class list</h2>
13     <div class="card-stacked">
14         <form action="AddclassServlet" method="POST">
15             <span class="card-title" style="margin-right: 40px;">Class name</span>
16             <input type="text" name="classname" placeholder="class name"/><br/><br/>
17             <span class="card-title">Class Description</span>
18             <input type="text" name="description" placeholder="class description"/><br/><br/>
19             <div class="card-action">
20                 <input type="submit" class="fadeIn fourth" value="Submit"/>
21             </div>
22         </form>
23     </div>
24 </div>
25 </body>
26 </html>
```

addSubject.jsp

The screenshot shows the Eclipse IDE interface with the project 'LearnersAcademy' open. The 'Project Explorer' view on the left lists various Java files and web resources. The 'addSubject.jsp' file is selected and displayed in the central editor area. The code is a JSP page with some CSS classes defined. The status bar at the bottom shows the file is 'Writable' and the current time is 21:11:885.

```
1 <%@page import="database.LearnerSubjectDB"%>
2 <%@page import="entity.Subject"%>
3 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"%>
4 <%@page encoding="ISO-8859-1"%>
5 <%@include file="include/navigation.jsp"%>
6 <!DOCTYPE html>
7 <html>
8 <head>
9 <meta charset="ISO-8859-1">
10 <title>Insert title here</title>
11 </head>
12 <body>
13 <div class="container" style="margin-top: 20px; margin-left: 50px;">
14 <h2 class="header">Add a Subject to master list</h2>
15 <div class="card-stacked">
16 <form action="AddSubjectServlet" method="POST">
17 <span class="card-title">Subject name</span>
18 <input type="text" name="Subjectname" placeholder="subject name"/><br/><br/>
19 <div class="card-action"><br/>
20 <input type="submit" class="fadeIn fourth" value="Submit" />
21 </div>
22 </form>
23 </div>
24 </div>
25 </body>
26 </html>
```

addTeacher.jsp

The screenshot shows the Eclipse IDE interface with the project 'LearnersAcademy' open. The 'Project Explorer' view on the left lists various Java files and web resources. The 'addTeacher.jsp' file is selected and displayed in the central editor area. The code is a JSP page with some CSS classes defined. The status bar at the bottom shows the file is 'Writable' and the current time is 28:19:1523.

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"%>
2 <%@page encoding="ISO-8859-1"%>
3 <%@include file="include/navigation.jsp"%>
4 <!DOCTYPE HTML>
5 <html>
6 <head>
7 <meta charset="ISO-8859-1">
8 <title>Insert title here</title>
9 </head>
10 <body>
11 <div class="container" style="margin-top: 20px; margin-left: 50px;">
12 <h2 class="header">Add a teacher to teacher list</h2>
13 <div class="card-stacked">
14 <form action="AddTeacherServlet" method="POST">
15 <span class="card-title" style="margin-right:35px;">First name</span>
16 <input type="text" name="firstname" placeholder="firstname"/><br/><br/>
17 <span class="card-title" style="margin-right:35px;">Last name</span>
18 <input type="text" name="lastname" placeholder="lastname"/><br/><br/>
19 <span class="card-title" style="margin-right:70px;">Email</span>
20 <input type="text" name="email" placeholder="email"/><br/><br/>
21 <span class="card-title">Phone Number</span>
22 <input type="text" name="phonenumber" placeholder="phone number"/><br/><br/>
23 <div class="card-action"><br/>
24 <input type="submit" class="fadeIn fourth" value="Submit" />
25 <input type="hidden" name="action" value="action1" />
26 </div>
27 </form>
28 </div>
29 </div>
30 </body>
31 </html>
```

assignClass.jsp

assignClass.jsp

```

eclipse-workspace - LearnersAcademy/src/main/webapp/assignClass.jsp - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer < PHP > assignClass.jsp
  TeacherClassServlet.java
  ViewStudentServlet.java
  ViewSubjectServlet.java
  ViewTeacherServlet.java
  > webapp
    > include
      navigation.jsp
    > META-INF
    > WEB-INF
      addClass.jsp
      addSubject.jsp
      addTeacher.jsp
      assignClass.jsp
      assignTeacher.jsp
      classReport.jsp
      dashboard.jsp
      index.jsp
      learning_unsplash.jpg
      subjects.jsp
      teachers.jsp
      viewClasses.jsp
      viewStudents.jsp
  > LockedMe.com [roseline.mills master]
  > Phase1JavaDemo [roseline.mills master]
  > Servers [roseline.mills master]
  > ServletDemo [roseline.mills master]
    Deployment Descriptor: ServletDemo
    JAX-WS Web Services
    Java Resources
    build
    src
  Writable Smart Insert 19 : 9 : 684
  1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
  2   pageEncoding="ISO-8859-1"%>
  3 <%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
  4 <%@include file="include/navigation.jsp"%>
  5 <!DOCTYPE html>
  6 <html>
  7   <head>
  8     <meta charset="ISO-8859-1">
  9     <title>Insert title here</title>
 10   </head>
 11   <body>
 12     <br/>
 13     <div class="container" style="margin-top: 20px; margin-left: 50px;">
 14       <h2 class="header">Assign class to Subject</h2>
 15       <form action="ClassSubjectServlet" method="POST">
 16         <span class="card-title">Select class name</span>
 17         <select name="class">
 18           <c:forEach var="learnerClass" items="${requestScope.classes}">
 19             <option value=" ${learnerClass.class_id} "> ${learnerClass.class_name}</option>
 20           </c:forEach>
 21         </select>
 22         <span class="card-title">Select subject name</span>
 23         <select name="subject">
 24           <c:forEach var="learnerSubject" items="${requestScope.subjects}">
 25             <option value=" ${learnerSubject.subject_id} "> ${learnerSubject.subject_name}</option>
 26           </c:forEach>
 27         </select>
 28         <div class="card-action"><br/>
 29           <input type="submit" class="fadeIn fourth" value="Submit" /></div>
 30       </form>
 31     </div>
 32   </body>
 33 </html>
 11:27 PM 3/11/2023

```

assignTeacher.jsp

assignTeacher.jsp

```

eclipse-workspace - LearnersAcademy/src/main/webapp/assignTeacher.jsp - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer < PHP > assignTeacher.jsp
  TeacherClassServlet.java
  ViewStudentServlet.java
  ViewSubjectServlet.java
  ViewTeacherServlet.java
  > webapp
    > include
      navigation.jsp
    > META-INF
    > WEB-INF
      addClass.jsp
      addSubject.jsp
      addTeacher.jsp
      assignClass.jsp
      assignTeacher.jsp
      classReport.jsp
      dashboard.jsp
      index.jsp
      learning_unsplash.jpg
      subjects.jsp
      teachers.jsp
      viewClasses.jsp
      viewStudents.jsp
  > LockedMe.com [roseline.mills master]
  > Phase1JavaDemo [roseline.mills master]
  > Servers [roseline.mills master]
  > ServletDemo [roseline.mills master]
    Deployment Descriptor: ServletDemo
    JAX-WS Web Services
    Java Resources
    build
    src
  Writable Smart Insert 23 : 13 : 879
  1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
  2   pageEncoding="ISO-8859-1"%>
  3 <%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
  4 <%@include file="include/navigation.jsp"%>
  5 <!DOCTYPE html>
  6 <html>
  7   <head>
  8     <meta charset="ISO-8859-1">
  9     <title>Assign teacher to class</title>
 10   </head>
 11   <body>
 12     <br/>
 13     <div class="container" style="margin-top: 20px; margin-left: 50px;">
 14       <h2 class="header">Assign teacher to class</h2><br/>
 15       <form action="TeacherClassServlet" method="POST">
 16         <span class="card-title">Select teacher name</span>
 17         <select name="teacher">
 18           <c:forEach var="teacher" items="${requestScope.teachers}">
 19             <option value=" ${teacher.teacher_id} "> ${teacher.first_name} ${teacher.last_name}</option>
 20           </c:forEach>
 21         </select>
 22         <span class="card-title">Select class name</span>
 23         <select name="class">
 24           <c:forEach var="learnerClass" items="${requestScope.classes}">
 25             <option value=" ${learnerClass.class_id} "> ${learnerClass.class_name}</option>
 26           </c:forEach>
 27         </select>
 28         <div class="card-action"><br/>
 29           <input type="submit" class="fadeIn fourth" value="Submit" /></div>
 30       </form>
 31     </div>
 32   </body>
 33 </html>
 11:28 PM 3/11/2023

```

classReport.jsp

eclipse-workspace - LearnersAcademy/src/main/webapp/classReport.jsp - Eclipse IDE

```

1 <%@ page language="Java" contentType="text/html; charset=ISO-8859-1"
2     pageEncoding="ISO-8859-1"%>
3 <%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
4 <%@include file="include/navigation.jsp"%>
5 <!DOCTYPE html>
6 <html>
7 <head>
8     <meta charset="ISO-8859-1">
9     <title>Class Report</title>
10    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
11          rel="stylesheet" integrity="sha384-EV9zT4AUQ6JzVXnZDChz/0UxOc9vLWZjDQD0Tqo9eR1Ovwq6Q=" crossorigin="anonymous">
12 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLABNl+NtUVF
13         crossorigin="anonymous">
14 </script>
15 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLABNl+NtUVF
16         crossorigin="anonymous">
17 </script>
18 <body>
19 <div class="container">
20     <h2>Class Report</h2>
21     <br/>
22     <table class="table">
23         <tr>
24             <th>class_id</th>
25             <th>class_name</th>
26             <th>teacher_assigned</th>
27             <th>subject_name</th>
28             <th>teacher assigned to</th>
29             <th>students assigned</th>
30         </tr>
31         <c:forEach var="classReport" items="${requestScope.classreports}">
32             <tr>
33                 <td>${classReport.class_id}</td>
34                 <td>${classReport.class_name}</td>
35                 <td>${classReport.class_description}</td>
36                 <td>${classReport.subject_name}</td>
37                 <td>${classReport.teacher_fname} ${classReport.teacher_lname}</td>
38                 <td>${classReport.student_fname} ${classReport.student_lname}</td>
39             </tr>
40         </c:forEach>
41     </table>
42 </div>
43 </body>
44 </html>

```

Writable | Smart Insert | 24 : 30 : 864 | 11:29 PM 3/11/2023

Dashboard.jsp

eclipse-workspace - LearnersAcademy/src/main/webapp/dashboard.jsp - Eclipse IDE

```

1 <%@ page language="Java" contentType="text/html; charset=ISO-8859-1"
2     pageEncoding="ISO-8859-1"%>
3 <%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
4 <%@include file="include/navigation.jsp"%>
5 <!DOCTYPE html>
6 <html>
7     <head>
8         <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
9              rel="stylesheet" integrity="sha384-EV9zT4AUQ6JzVXnZDChz/0UxOc9vLWZjDQD0Tqo9eR1Ovwq6Q=" crossorigin="anonymous">
10        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EV9zT4AUQ6JzVXnZDChz/0UxOc9vLWZjDQD0Tqo9eR1Ovwq6Q=" crossorigin="anonymous">
11        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLABNl+NtUVF
12             crossorigin="anonymous">
13 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLABNl+NtUVF
14             crossorigin="anonymous">
15 <style>
16     .titleAdministrator dashboard{title}
17 </style>
18 <body>
19     <nav class="navbar navbar-expand-lg navbar-light bg-light">
20         <div class="container-fluid">
21             <a class="navbar-brand" style="margin-left: 15px;" href="#">Learner's Academy Dashboard</a>
22             <div class="collapse navbar-collapse" id="navSupportedContent">
23                 <ul class="navbar-nav me-auto mb-2 mb-lg-0">
24                     <li class="nav-item"><a class="nav-link active" aria-current="page" href="index.jsp">Logout</a></li>
25                 </ul>
26             </div>
27         </div>
28     </nav>
29     <div class="nav-section" style="margin-top: 20px; margin-left: 50px;">
30         <ul class="nav nav-nou_me-auto mb-2 mb-lg-0">
31             <li class="nav-item"><a href="#">Set up master list of subject</a></li>
32             <li class="nav-item"><a href="#">Set up master list of teachers</a></li>
33             <li class="nav-item"><a href="#">Set up master list of classes</a></li>
34             <li class="nav-item"><a href="#">Assign classes for subjects</a></li>
35             <li class="nav-item"><a href="#">Assign teachers to a class</a></li>
36             <li class="nav-item"><a href="#">Get master list of students</a></li>
37             <li class="nav-item"><a href="#">View class report</a></li>
38         </ul>
39     </div>
40 </body>
41 </html>

```

Writable | Smart Insert | 28 : 15 : 1340 | 11:29 PM 3/11/2023

Index.jsp

eclipse-workspace - LearnersAcademy/src/main/webapp/index.jsp - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

```

    TeacherClassServlet.java
    ViewStudentServlet.java
    ViewSubjectServlet.java
    ViewTeacherServlet.java
    > webapp
        > include
            navigation.jsp
        > META-INF
        > WEB-INF
            addClass.jsp
            addSubject.jsp
            addTeacher.jsp
            assignClass.jsp
            assignTeacher.jsp
            classReport.jsp
            dashboard.jsp
            index.jsp
            learning_unsplash.jpg
            subjects.jsp
            teachers.jsp
            viewClasses.jsp
            viewStudents.jsp
    > LockedMe.com [roseline.mills master]
    > Phase1JavaDemo [roseline.mills master]
    > Servers [roseline.mills master]
    > ServletDemo [roseline.mills master]
        Deployment Descriptor: ServletDemo
        JAX-WS Web Services
        Java Resources
        build
        > src

```

index.jsp

```

1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2     pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6     <meta charset="ISO-8859-1">
7     <title>Insert title here</title>
8     </head>
9     <body>
10    <div class="container">
11        <div class="col s12 m7">
12            <h2 class="header">Learner's Academy School System</h2>
13            <div class="card horizontal">
14                <div class="card-image">
15                    
16                </div>
17                <div class="card-stacked">
18                    <form action="AdminLoginServlet" method="POST">
19                        <span class="card-title">Username</span>
20                        <input type="text" name="username" placeholder="username" value="rose"/><br/><br/>
21                        <span class="card-title">Password</span>
22                        <input type="password" name="password" placeholder="password" value="rose123"/><br/>
23                        <div class="card-action">
24                            <input type="submit" class="fadeIn fourth" value="Log In" />
25                        </div>
26                    </form>
27                </div>
28            </div>
29        </div>
30    </div>
31    </body>
32 </html>

```

Writable Smart Insert 29 : 11 : 1194 11:30 PM 3/11/2023

subjects.jsp

eclipse-workspace - LearnersAcademy/src/main/webapp/subjects.jsp - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

```

    TeacherClassServlet.java
    ViewStudentServlet.java
    ViewSubjectServlet.java
    ViewTeacherServlet.java
    > webapp
        > include
            navigation.jsp
        > META-INF
        > WEB-INF
            addClass.jsp
            addSubject.jsp
            addTeacher.jsp
            assignClass.jsp
            assignTeacher.jsp
            classReport.jsp
            dashboard.jsp
            index.jsp
            learning_unsplash.jpg
            subjects.jsp
            teachers.jsp
            viewClasses.jsp
            viewStudents.jsp
    > LockedMe.com [roseline.mills master]
    > Phase1JavaDemo [roseline.mills master]
    > Servers [roseline.mills master]
    > ServletDemo [roseline.mills master]
        Deployment Descriptor: ServletDemo
        JAX-WS Web Services
        Java Resources
        build
        > src

```

subjects.jsp

```

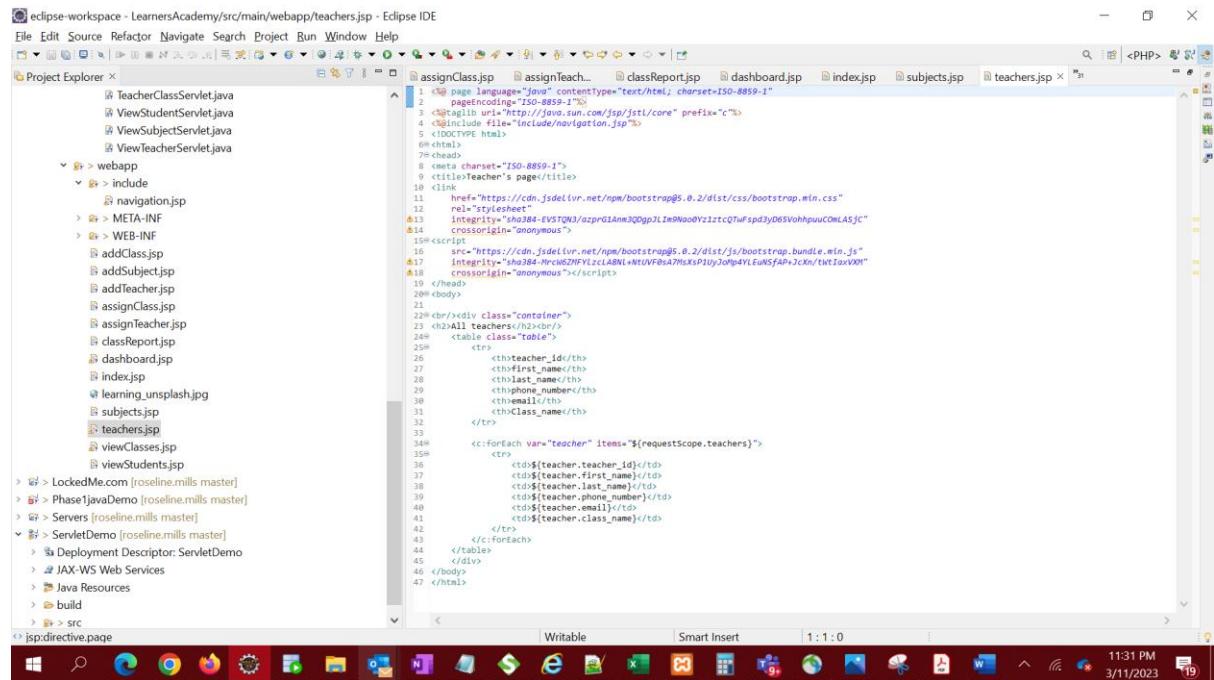
1 <%@ page language="Java" contentType="text/html; charset=ISO-8859-1"
2     pageEncoding="ISO-8859-1"%>
3 <@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
4 <@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
5 <!DOCTYPE html>
6 <html>
7 <head>
8     <meta charset="ISO-8859-1">
9     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
10         rel="stylesheet">
11     <link href="https://azprG1Anm3Q0gpJLIm9Noo0Yz1ztCQThFspd3yO6SVohpuuCoLASjC"
12         crossorigin="anonymous">
13     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQ03/azprG1Anm3Q0gpJLIm9Noo0Yz1ztCQThFspd3yO6SVohpuuCoLASjC"
14     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcWEZMFLzclABNL+NTUVF0sA7HnsXsP10yJ0NgYLEuNsFAP+jCkNnTwIxVX0M"
15         crossorigin="anonymous"></script>
16 </head>
17 <body>
18     <title>List of all subjects</title>
19     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
20         rel="stylesheet">
21     <link href="https://azprG1Anm3Q0gpJLIm9Noo0Yz1ztCQThFspd3yO6SVohpuuCoLASjC"
22         crossorigin="anonymous">
23     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcWEZMFLzclABNL+NTUVF0sA7HnsXsP10yJ0NgYLEuNsFAP+jCkNnTwIxVX0M"
24         crossorigin="anonymous"></script>
25 </body>
26 </html>

```

html/body/div/table/cforeach/t#:text

Writable Smart Insert 39 : 47 : 1724 11:30 PM 3/11/2023

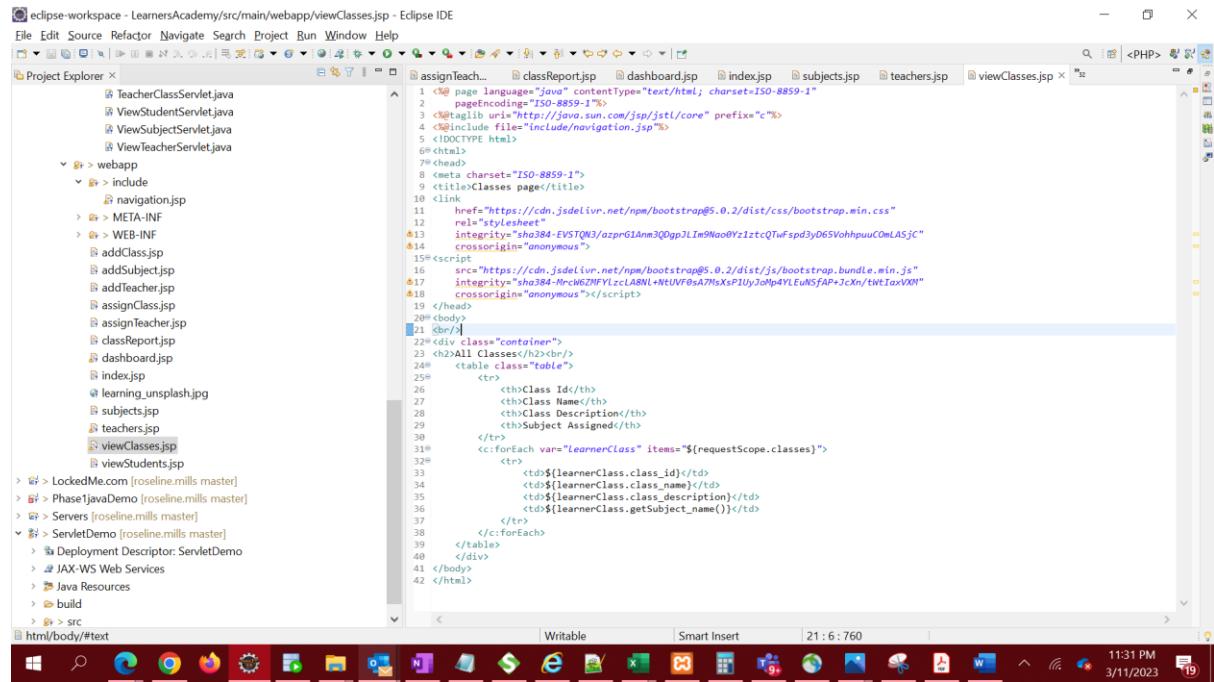
teachers.jsp



The screenshot shows the Eclipse IDE interface with the 'teachers.jsp' file open in the editor. The code is a JSP page with embedded Java code. It includes imports for page directives, taglibs, and navigation.jsp. The page has a head section with meta tags and links to Bootstrap CSS and JS files. The body contains a table to display teacher information.

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2     pageEncoding="ISO-8859-1"%>
3 <%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
4 <%@include file="include/navigation.jsp"%>
5 <%@page type="text/html" %>
6 <%@page errorPage="error.jsp" %>
7 <%@page session="true" %>
8 <%@page buffer="none" %>
9 <%@page isThreadSafe="true" %>
10 <%@page trimDirectiveWhitespaces="true" %>
11 <meta charset="ISO-8859-1">
12 <title>Teacher's page</title>
13 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3Q0gpJLIm9fYzW3zKlqJl8gH1o+Zk58KF+iD8bJW3q5" crossorigin="anonymous">
14 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcWfirhpuqz9fetjwHvH08t3yVdElw+0z1xJWZJvP" crossorigin="anonymous"></script>
15 <head>
16 </head>
17 <body>
18 <div class="container">
19 <h2>All teachers:</h2>
20 <table class="table">
21 <thead>
22 <tr>
23 <th>Teacher_id</th>
24 <th>First_name</th>
25 <th>Last_name</th>
26 <th>Phone number</th>
27 <th>Email</th>
28 <th>Class_name</th>
29 </tr>
30 <c:forEach var="teacher" items="${requestScope.teachers}">
31 <tr>
32 <td>${teacher.teacher_id}</td>
33 <td>${teacher.first_name}</td>
34 <td>${teacher.last_name}</td>
35 <td>${teacher.phone}</td>
36 <td>${teacher.email}</td>
37 <td>${teacher.class_name}</td>
38 </tr>
39 </c:forEach>
40 </table>
41 </div>
42 </body>
43 </html>
```

viewClasses.jsp



The screenshot shows the Eclipse IDE interface with the 'viewClasses.jsp' file open in the editor. The code is a JSP page with embedded Java code. It includes imports for page directives, taglibs, and navigation.jsp. The page has a head section with meta tags and links to Bootstrap CSS and JS files. The body contains a table to display learner class information.

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2     pageEncoding="ISO-8859-1"%>
3 <%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
4 <%@include file="include/navigation.jsp"%>
5 <%@DOCTYPE html%>
6 <%@page type="text/html" %>
7 <%@page errorPage="error.jsp" %>
8 <%@page session="true" %>
9 <%@page buffer="none" %>
10 <%@page isThreadSafe="true" %>
11 <%@page trimDirectiveWhitespaces="true" %>
12 <meta charset="ISO-8859-1">
13 <title>Classes page</title>
14 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3Q0gpJLIm9fYzW3zKlqJl8gH1o+Zk58KF+iD8bJW3q5" crossorigin="anonymous">
15 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcWfirhpuqz9fetjwHvH08t3yVdElw+0z1xJWZJvP" crossorigin="anonymous"></script>
16 <head>
17 </head>
18 <body>
19 <div class="container">
20 <h2>All Classes:</h2>
21 <table class="table">
22 <thead>
23 <tr>
24 <th>Class Id</th>
25 <th>Class Name</th>
26 <th>Description</th>
27 <th>Subject Assigned</th>
28 </tr>
29 <c:forEach var="learnerClass" items="${requestScope.classes}">
30 <tr>
31 <td>${learnerClass.class_id}</td>
32 <td>${learnerClass.class_name}</td>
33 <td>${learnerClass.class_description}</td>
34 <td>${learnerClass.getSubject_name()}</td>
35 </tr>
36 </c:forEach>
37 </table>
38 </div>
39 </body>
40 </html>
```

viewStudents.jsp

eclipse-workspace - LearnersAcademy/src/main/webapp/viewStudents.jsp - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

- TeacherClassServlet.java
- ViewStudentServlet.java
- ViewSubjectServlet.java
- ViewTeacherServlet.java
- webapp
 - include
 - navigation.jsp
 - META-INF
 - WEB-INF
 - addClass.jsp
 - addSubject.jsp
 - addTeacher.jsp
 - assignClass.jsp
 - assignTeacher.jsp
 - classReport.jsp
 - dashboard.jsp
 - index.jsp
 - learning_unsplash.jpg
 - subjects.jsp
 - teachers.jsp
 - viewClasses.jsp
 - viewStudents.jsp
- LockedMe.com [roseline.mills master]
- Phase1JavaDemo [roseline.mills master]
- Servers [roseline.mills master]
- ServletDemo [roseline.mills master]
 - Deployment Descriptor: ServletDemo
 - JAX-WS Web Services
 - Java Resources
 - build
 - src

classReport.jsp dashboard.jsp index.jsp subjects.jsp teachers.jsp viewClasses.jsp *viewStudents.jsp

```
1 <%@page language="Java" contentType="text/html; charset=ISO-8859-1"
2 pageEncoding="ISO-8859-1"%>
3 <%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
4 <%@taglib uri="entity/student"%>
5 <%@include file="include/navigation.jsp"%>
6 <!DOCTYPE html>
7<html>
8 <head>
9 <meta charset="ISO-8859-1">
10 <title>Student's page</title>
11 <link
12 href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
13 rel="stylesheet"
14 integrity="sha384-EVSTQN3/azprG14aWjYHr3vRZJL/WzBLQgxrARqE9yvM+QD2iQZl93dK2o0OYV12tcQ7uFspd3yD65VohhpuCOMASjC"
15 crossorigin="anonymous">
16 <script
17 src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
18 integrity="sha384-MrcWfirhTpXyvJjuHtw7JzPd5q+D6Ri3/+PSwlvO6TxeX5hduu1l0A3I7fXqJ9S6B8O"
19 crossorigin="anonymous"></script>
20 </head>
21 <body>
22 <div>
23 <div class="container">
24 <h2>All Students</h2><br/>
25 <table class="table">
26 <thead>
27 <tr>
28 <th>student_id</th>
29 <th>first_name</th>
30 <th>last_name</th>
31 <th>phone_number</th>
32 <th>email</th>
33 <th>class</th>
34 </tr>
35 <c:forEach var="student" items="${requestScope.students}">
36 <tr>
37 <td>${student.student_id}</td>
38 <td>${student.first_name}</td>
39 <td>${student.last_name}</td>
40 <td>${student.phone_number}</td>
41 <td>${student.email}</td>
42 <td>${student.learner_class}</td>
43 </tr>
44 </c:forEach>
45 </table>
46 </div>
47 </body>
48 </html>
```

Writable Smart Insert 22:6:809

1132 PM 3/11/2023