

PL/SQL

1.CONTROL STRUCTURES

```
CREATE TABLE customers ( customer_id INT, name VARCHAR(50), age INT, balance DECIMAL(10,2), interest_rate DECIMAL(5,2), isvip VARCHAR(5) );
```

```
-- Insert some sample rows INSERT INTO customers VALUES (1, 'Alice', 65, 12000.00, 7.5, 'FALSE'); INSERT INTO customers VALUES (2, 'Bob', 45, 8000.00, 8.0, 'FALSE'); INSERT INTO customers VALUES (3, 'Charlie', 70, 15000.00, 6.5, 'FALSE'); INSERT INTO customers VALUES (4, 'David', 30, 5000.00, 9.0, 'FALSE');
```

```
-- Scenario 1: Apply 1% discount to customers above 60 years old UPDATE customers SET interest_rate = interest_rate - (interest_rate * 0.01) WHERE age > 60;
```

```
-- Scenario 2: Promote to VIP based on balance UPDATE customers SET isvip = 'TRUE' WHERE balance > 10000;
```

```
-- View updated table SELECT * FROM customers;
```

OUTPUT SCREENSHOT:

The screenshot shows the OneCompiler web interface. On the left, the code editor displays a file named 'queries.sql' containing the provided PL/SQL script. On the right, the 'Output' pane shows the results of the executed queries. The output table is as follows:

customer_id	name	age	balance
1	Alice	65	12000.00
2	Bob	45	8000.00
3	Charlie	70	15000.00
4	David	30	5000.00

2.ERROR HANDLING:

```
CREATE TABLE accounts (
```

```

account_id INT PRIMARY KEY,
balance DECIMAL(10,2)
);

CREATE TABLE employees (
    employee_id INT PRIMARY KEY,
    name VARCHAR(50),
    salary DECIMAL(10,2)
);

CREATE TABLE customers (
    customer_id INT PRIMARY KEY,
    name VARCHAR(50),
    balance DECIMAL(10,2)
);

CREATE TABLE error_log (
    message VARCHAR(255),
    error_time DATETIME
);

-- Step 2: Insert sample data
INSERT INTO accounts VALUES (1, 5000.00), (2, 3000.00);
INSERT INTO employees VALUES (101, 'Alice', 60000.00), (102, 'Bob', 50000.00);
INSERT INTO customers VALUES (201, 'Charlie', 7000.00);

-- Step 3: Transfer funds with error handling
BEGIN TRY
    BEGIN TRANSACTION;

    DECLARE @balance DECIMAL(10,2);
    DECLARE @amount DECIMAL(10,2) = 10000;
    DECLARE @from_acct INT = 2;
    DECLARE @to_acct INT = 1;

    SELECT @balance = balance FROM accounts WHERE account_id = @from_acct;

    IF @balance < @amount
        THROW 50001, 'Insufficient funds', 1;

```

```

UPDATE accounts SET balance = balance - @amount WHERE account_id =
@from_acct;
UPDATE accounts SET balance = balance + @amount WHERE account_id =
@to_acct;

COMMIT;
END TRY
BEGIN CATCH
ROLLBACK;
INSERT INTO error_log(message, error_time)
VALUES(ERROR_MESSAGE(), GETDATE());
END CATCH;

-- Step 4: Update salary with error handling
BEGIN TRY
BEGIN TRANSACTION;

DECLARE @emp_id INT = 999;
DECLARE @percent DECIMAL(5,2) = 10;

UPDATE employees
SET salary = salary + (salary * @percent / 100)
WHERE employee_id = @emp_id;

IF @@ROWCOUNT = 0
    THROW 50002, 'Employee not found', 1;

COMMIT;
END TRY
BEGIN CATCH
ROLLBACK;
INSERT INTO error_log(message, error_time)
VALUES(ERROR_MESSAGE(), GETDATE());
END CATCH;

-- Step 5: Add customer with duplicate ID to test error handling
BEGIN TRY
INSERT INTO customers(customer_id, name, balance)
VALUES(201, 'DuplicateCharlie', 9999.00);
END TRY
BEGIN CATCH

```

```

INSERT INTO error_log(message, error_time)
VALUES(ERROR_MESSAGE(), GETDATE());
END CATCH;

```

-- Step 6: View results

```

SELECT * FROM accounts;
SELECT * FROM employees;
SELECT * FROM customers;
SELECT * FROM error_log;

```

OUTPUT SCREENSHOT

```

1 -- create
2
3 -- CUSTOMER table
4 CREATE TABLE accounts (
5 account_id INT PRIMARY KEY,
6 balance DECIMAL(18,2)
7 );
8
9
10-- CREATE TABLE employees (
11 employee_id INT PRIMARY KEY,
12 name VARCHAR(50),
13 salary DECIMAL(18,2)
14 );
15
16-- CREATE TABLE customers (
17 customer_id INT PRIMARY KEY,
18 name VARCHAR(50),
19 balance DECIMAL(18,2)
20 );
21
22-- CREATE TABLE error_log (
23 message VARCHAR(255),
24 error_time DATETIME
25 );
26
27-- Step 2: Insert sample data
28 INSERT INTO accounts VALUES (1, 5000.00), (2, 3000.00);
29 INSERT INTO employees VALUES (101, 'Alice', 60000.00), (102, 'Bob', 50000.00);
30 INSERT INTO customers VALUES (201, 'Charlie', 7000.00);
31
32-- Step 3: Transfer funds with error handling
33 BEGIN TRY
34 BEGIN TRANSACTION;
35
36 DECLARE @balance DECIMAL(18,2);
37 DECLARE @amount DECIMAL(18,2) = 10000;
38 DECLARE @from_acct INT = 2;
39 DECLARE @to_acct INT = 1;
40
41 SELECT @balance = balance FROM accounts WHERE account_id = @from_acct;
42
43 IF @balance < @amount
44 BEGIN
45 RAISERROR('Insufficient funds', 16, 1);
46 END
47 ELSE
48 BEGIN
49 UPDATE accounts SET balance = balance - @amount WHERE account_id = @from_acct;
50
51 UPDATE accounts SET balance = balance + @amount WHERE account_id = @to_acct;
52 END
53 END TRANSACTION;
54
55 END TRY
56 BEGIN CATCH
57
58 IF @@TRANCOUNT > 0
59 BEGIN
60 ROLLBACK TRANSACTION;
61 END
62
63 DECLARE @error_message NVARCHAR(4000) = ERROR_MESSAGE();
64 DECLARE @error_severity INT = ERROR_SEVERITY();
65 DECLARE @error_state INT = ERROR_STATE();
66
67 RAISERROR(@error_message, @error_severity, @error_state);
68 END CATCH
69
70
71-- Step 4: Insert test data
72-- Step 5: View results
73
74-- Step 6: View results
75
76-- Step 7: Clean up
77
78-- Step 8: View results
79
80-- Step 9: View results
81
82-- Step 10: View results
83
84-- Step 11: View results
85
86-- Step 12: View results
87
88-- Step 13: View results
89
90-- Step 14: View results
91
92-- Step 15: View results
93
94-- Step 16: View results
95
96-- Step 17: View results
97
98-- Step 18: View results
99
100-- Step 19: View results
101
102-- Step 20: View results
103
104-- Step 21: View results
105
106-- Step 22: View results
107
108-- Step 23: View results
109
110-- Step 24: View results
111
112-- Step 25: View results
113
114-- Step 26: View results
115
116-- Step 27: View results
117
118-- Step 28: View results
119
120-- Step 29: View results
121
122-- Step 30: View results
123
124-- Step 31: View results
125
126-- Step 32: View results
127
128-- Step 33: View results
129
130-- Step 34: View results
131
132-- Step 35: View results
133
134-- Step 36: View results
135
136-- Step 37: View results
137
138-- Step 38: View results
139
140-- Step 39: View results
141
142-- Step 40: View results
143
144-- Step 41: View results
145
146-- Step 42: View results
147
148-- Step 43: View results
149
150-- Step 44: View results
151
152-- Step 45: View results
153
154-- Step 46: View results
155
156-- Step 47: View results
157
158-- Step 48: View results
159
160-- Step 49: View results
161
162-- Step 50: View results
163
164-- Step 51: View results
165
166-- Step 52: View results
167
168-- Step 53: View results
169
170-- Step 54: View results
171
172-- Step 55: View results
173
174-- Step 56: View results
175
176-- Step 57: View results
177
178-- Step 58: View results
179
180-- Step 59: View results
181
182-- Step 60: View results
183
184-- Step 61: View results
185
186-- Step 62: View results
187
188-- Step 63: View results
189
190-- Step 64: View results
191
192-- Step 65: View results
193
194-- Step 66: View results
195
196-- Step 67: View results
197
198-- Step 68: View results
199
200-- Step 69: View results
201
202-- Step 70: View results
203
204-- Step 71: View results
205
206-- Step 72: View results
207
208-- Step 73: View results
209
210-- Step 74: View results
211
212-- Step 75: View results
213
214-- Step 76: View results
215
216-- Step 77: View results
217
218-- Step 78: View results
219
220-- Step 79: View results
221
222-- Step 80: View results
223
224-- Step 81: View results
225
226-- Step 82: View results
227
228-- Step 83: View results
229
230-- Step 84: View results
231
232-- Step 85: View results
233
234-- Step 86: View results
235
236-- Step 87: View results
237
238-- Step 88: View results
239
240-- Step 89: View results
241
242-- Step 90: View results
243
244-- Step 91: View results
245
246-- Step 92: View results
247
248-- Step 93: View results
249
250-- Step 94: View results
251
252-- Step 95: View results
253
254-- Step 96: View results
255
256-- Step 97: View results
257
258-- Step 98: View results
259
260-- Step 99: View results
261
262-- Step 100: View results
263
264-- Step 101: View results
265
266-- Step 102: View results
267
268-- Step 103: View results
269
270-- Step 104: View results
271
272-- Step 105: View results
273
274-- Step 106: View results
275
276-- Step 107: View results
277
278-- Step 108: View results
279
280-- Step 109: View results
281
282-- Step 110: View results
283
284-- Step 111: View results
285
286-- Step 112: View results
287
288-- Step 113: View results
289
290-- Step 114: View results
291
292-- Step 115: View results
293
294-- Step 116: View results
295
296-- Step 117: View results
297
298-- Step 118: View results
299
300-- Step 119: View results
301
302-- Step 120: View results
303
304-- Step 121: View results
305
306-- Step 122: View results
307
308-- Step 123: View results
309
310-- Step 124: View results
311
312-- Step 125: View results
313
314-- Step 126: View results
315
316-- Step 127: View results
317
318-- Step 128: View results
319
320-- Step 129: View results
321
322-- Step 130: View results
323
324-- Step 131: View results
325
326-- Step 132: View results
327
328-- Step 133: View results
329
330-- Step 134: View results
331
332-- Step 135: View results
333
334-- Step 136: View results
335
336-- Step 137: View results
337
338-- Step 138: View results
339
340-- Step 139: View results
341
342-- Step 140: View results
343
344-- Step 141: View results
345
346-- Step 142: View results
347
348-- Step 143: View results
349
350-- Step 144: View results
351
352-- Step 145: View results
353
354-- Step 146: View results
355
356-- Step 147: View results
357
358-- Step 148: View results
359
360-- Step 149: View results
361
362-- Step 150: View results
363
364-- Step 151: View results
365
366-- Step 152: View results
367
368-- Step 153: View results
369
370-- Step 154: View results
371
372-- Step 155: View results
373
374-- Step 156: View results
375
376-- Step 157: View results
377
378-- Step 158: View results
379
380-- Step 159: View results
381
382-- Step 160: View results
383
384-- Step 161: View results
385
386-- Step 162: View results
387
388-- Step 163: View results
389
390-- Step 164: View results
391
392-- Step 165: View results
393
394-- Step 166: View results
395
396-- Step 167: View results
397
398-- Step 168: View results
399
400-- Step 169: View results
401
402-- Step 170: View results
403
404-- Step 171: View results
405
406-- Step 172: View results
407
408-- Step 173: View results
409
410-- Step 174: View results
411
412-- Step 175: View results
413
414-- Step 176: View results
415
416-- Step 177: View results
417
418-- Step 178: View results
419
420-- Step 179: View results
421
422-- Step 180: View results
423
424-- Step 181: View results
425
426-- Step 182: View results
427
428-- Step 183: View results
429
430-- Step 184: View results
431
432-- Step 185: View results
433
434-- Step 186: View results
435
436-- Step 187: View results
437
438-- Step 188: View results
439
440-- Step 189: View results
441
442-- Step 190: View results
443
444-- Step 191: View results
445
446-- Step 192: View results
447
448-- Step 193: View results
449
450-- Step 194: View results
451
452-- Step 195: View results
453
454-- Step 196: View results
455
456-- Step 197: View results
457
458-- Step 198: View results
459
460-- Step 199: View results
461
462-- Step 200: View results
463
464-- Step 201: View results
465
466-- Step 202: View results
467
468-- Step 203: View results
469
470-- Step 204: View results
471
472-- Step 205: View results
473
474-- Step 206: View results
475
476-- Step 207: View results
477
478-- Step 208: View results
479
480-- Step 209: View results
481
482-- Step 210: View results
483
484-- Step 211: View results
485
486-- Step 212: View results
487
488-- Step 213: View results
489
490-- Step 214: View results
491
492-- Step 215: View results
493
494-- Step 216: View results
495
496-- Step 217: View results
497
498-- Step 218: View results
499
500-- Step 219: View results
501
502-- Step 220: View results
503
504-- Step 221: View results
505
506-- Step 222: View results
507
508-- Step 223: View results
509
510-- Step 224: View results
511
512-- Step 225: View results
513
514-- Step 226: View results
515
516-- Step 227: View results
517
518-- Step 228: View results
519
520-- Step 229: View results
521
522-- Step 230: View results
523
524-- Step 231: View results
525
526-- Step 232: View results
527
528-- Step 233: View results
529
530-- Step 234: View results
531
532-- Step 235: View results
533
534-- Step 236: View results
535
536-- Step 237: View results
537
538-- Step 238: View results
539
540-- Step 239: View results
541
542-- Step 240: View results
543
544-- Step 241: View results
545
546-- Step 242: View results
547
548-- Step 243: View results
549
550-- Step 244: View results
551
552-- Step 245: View results
553
554-- Step 246: View results
555
556-- Step 247: View results
557
558-- Step 248: View results
559
560-- Step 249: View results
561
562-- Step 250: View results
563
564-- Step 251: View results
565
566-- Step 252: View results
567
568-- Step 253: View results
569
570-- Step 254: View results
571
572-- Step 255: View results
573
574-- Step 256: View results
575
576-- Step 257: View results
577
578-- Step 258: View results
579
580-- Step 259: View results
581
582-- Step 260: View results
583
584-- Step 261: View results
585
586-- Step 262: View results
587
588-- Step 263: View results
589
590-- Step 264: View results
591
592-- Step 265: View results
593
594-- Step 266: View results
595
596-- Step 267: View results
597
598-- Step 268: View results
599
600-- Step 269: View results
601
602-- Step 270: View results
603
604-- Step 271: View results
605
606-- Step 272: View results
607
608-- Step 273: View results
609
610-- Step 274: View results
611
612-- Step 275: View results
613
614-- Step 276: View results
615
616-- Step 277: View results
617
618-- Step 278: View results
619
620-- Step 279: View results
621
622-- Step 280: View results
623
624-- Step 281: View results
625
626-- Step 282: View results
627
628-- Step 283: View results
629
630-- Step 284: View results
631
632-- Step 285: View results
633
634-- Step 286: View results
635
636-- Step 287: View results
637
638-- Step 288: View results
639
640-- Step 289: View results
641
642-- Step 290: View results
643
644-- Step 291: View results
645
646-- Step 292: View results
647
648-- Step 293: View results
649
650-- Step 294: View results
651
652-- Step 295: View results
653
654-- Step 296: View results
655
656-- Step 297: View results
657
658-- Step 298: View results
659
660-- Step 299: View results
661
662-- Step 300: View results
663
664-- Step 301: View results
665
666-- Step 302: View results
667
668-- Step 303: View results
669
670-- Step 304: View results
671
672-- Step 305: View results
673
674-- Step 306: View results
675
676-- Step 307: View results
677
678-- Step 308: View results
679
680-- Step 309: View results
681
682-- Step 310: View results
683
684-- Step 311: View results
685
686-- Step 312: View results
687
688-- Step 313: View results
689
690-- Step 314: View results
691
692-- Step 315: View results
693
694-- Step 316: View results
695
696-- Step 317: View results
697
698-- Step 318: View results
699
700-- Step 319: View results
701
702-- Step 320: View results
703
704-- Step 321: View results
705
706-- Step 322: View results
707
708-- Step 323: View results
709
710-- Step 324: View results
711
712-- Step 325: View results
713
714-- Step 326: View results
715
716-- Step 327: View results
717
718-- Step 328: View results
719
720-- Step 329: View results
721
722-- Step 330: View results
723
724-- Step 331: View results
725
726-- Step 332: View results
727
728-- Step 333: View results
729
730-- Step 334: View results
731
732-- Step 335: View results
733
734-- Step 336: View results
735
736-- Step 337: View results
737
738-- Step 338: View results
739
740-- Step 339: View results
741
742-- Step 340: View results
743
744-- Step 341: View results
745
746-- Step 342: View results
747
748-- Step 343: View results
749
750-- Step 344: View results
751
752-- Step 345: View results
753
754-- Step 346: View results
755
756-- Step 347: View results
757
758-- Step 348: View results
759
760-- Step 349: View results
761
762-- Step 350: View results
763
764-- Step 351: View results
765
766-- Step 352: View results
767
768-- Step 353: View results
769
770-- Step 354: View results
771
772-- Step 355: View results
773
774-- Step 356: View results
775
776-- Step 357: View results
777
778-- Step 358: View results
779
780-- Step 359: View results
781
782-- Step 360: View results
783
784-- Step 361: View results
785
786-- Step 362: View results
787
788-- Step 363: View results
789
790-- Step 364: View results
791
792-- Step 365: View results
793
794-- Step 366: View results
795
796-- Step 367: View results
797
798-- Step 368: View results
799
800-- Step 369: View results
801
802-- Step 370: View results
803
804-- Step 371: View results
805
806-- Step 372: View results
807
808-- Step 373: View results
809
810-- Step 374: View results
811
812-- Step 375: View results
813
814-- Step 376: View results
815
816-- Step 377: View results
817
818-- Step 378: View results
819
820-- Step 379: View results
821
822-- Step 380: View results
823
824-- Step 381: View results
825
826-- Step 382: View results
827
828-- Step 383: View results
829
830-- Step 384: View results
831
832-- Step 385: View results
833
834-- Step 386: View results
835
836-- Step 387: View results
837
838-- Step 388: View results
839
840-- Step 389: View results
841
842-- Step 390: View results
843
844-- Step 391: View results
845
846-- Step 392: View results
847
848-- Step 393: View results
849
850-- Step 394: View results
851
852-- Step 395: View results
853
854-- Step 396: View results
855
856-- Step 397: View results
857
858-- Step 398: View results
859
860-- Step 399: View results
861
862-- Step 400: View results
863
864-- Step 401: View results
865
866-- Step 402: View results
867
868-- Step 403: View results
869
870-- Step 404: View results
871
872-- Step 405: View results
873
874-- Step 406: View results
875
876-- Step 407: View results
877
878-- Step 408: View results
879
880-- Step 409: View results
881
882-- Step 410: View results
883
884-- Step 411: View results
885
886-- Step 412: View results
887
888-- Step 413: View results
889
890-- Step 414: View results
891
892-- Step 415: View results
893
894-- Step 416: View results
895
896-- Step 417: View results
897
898-- Step 418: View results
899
900-- Step 419: View results
901
902-- Step 420: View results
903
904-- Step 421: View results
905
906-- Step 422: View results
907
908-- Step 423: View results
909
910-- Step 424: View results
911
912-- Step 425: View results
913
914-- Step 426: View results
915
916-- Step 427: View results
917
918-- Step 428: View results
919
920-- Step 429: View results
921
922-- Step 430: View results
923
924-- Step 431: View results
925
926-- Step 432: View results
927
928-- Step 433: View results
929
930-- Step 434: View results
931
932-- Step 435: View results
933
934-- Step 436: View results
935
936-- Step 437: View results
937
938-- Step 438: View results
939
940-- Step 439: View results
941
942-- Step 440: View results
943
944-- Step 441: View results
945
946-- Step 442: View results
947
948-- Step 443: View results
949
950-- Step 444: View results
951
952-- Step 445: View results
953
954-- Step 446: View results
955
956-- Step 447: View results
957
958-- Step 448: View results
959
960-- Step 449: View results
961
962-- Step 450: View results
963
964-- Step 451: View results
965
966-- Step 452: View results
967
968-- Step 453: View results
969
970-- Step 454: View results
971
972-- Step 455: View results
973
974-- Step 456: View results
975
976-- Step 457: View results
977
978-- Step 458: View results
979
980-- Step 459: View results
981
982-- Step 460: View results
983
984-- Step 461: View results
985
986-- Step 462: View results
987
988-- Step 463: View results
989
990-- Step 464: View results
991
992-- Step 465: View results
993
994-- Step 466: View results
995
996-- Step 467: View results
997
998-- Step 468: View results
999
1000-- Step 469: View results
1001
1002-- Step 470: View results
1003
1004-- Step 471: View results
1005
1006-- Step 472: View results
1007
1008-- Step 473: View results
1009
1010-- Step 474: View results
1011
1012-- Step 475: View results
1013
1014-- Step 476: View results
1015
1016-- Step 477: View results
1017
1018-- Step 478: View results
1019
1020-- Step 479: View results
1021
1022-- Step 480: View results
1023
1024-- Step 481: View results
1025
1026-- Step 482: View results
1027
1028-- Step 483: View results
1029
1030-- Step 484: View results
1031
1032-- Step 485: View results
1033
1034-- Step 486: View results
1035
1036-- Step 487: View results
1037
1038-- Step 488: View results
1039
1040-- Step 489: View results
1041
1042-- Step 490: View results
1043
1044-- Step 491: View results
1045
1046-- Step 492: View results
1047
1048-- Step 493: View results
1049
1050-- Step 494: View results
1051
1052-- Step 495: View results
1053
1054-- Step 496: View results
1055
1056-- Step 497: View results
1057
1058-- Step 498: View results
1059
1060-- Step 499: View results
1061
1062-- Step 500: View results
1063
1064-- Step 501: View results
1065
1066-- Step 502: View results
1067
1068-- Step 503: View results
1069
1070-- Step 504: View results
1071
1072-- Step 505: View results
1073
1074-- Step 506: View results
1075
1076-- Step 507: View results
1077
1078-- Step 508: View results
1079
1080-- Step 509: View results
1081
1082-- Step 510: View results
1083
1084-- Step 511: View results
1085
1086-- Step 512: View results
1087
1088-- Step 513: View results
1089
1090-- Step 514: View results
1091
1092-- Step 515: View results
1093
1094-- Step 516: View results
1095
1096-- Step 517: View results
1097
1098-- Step 518: View results
1099
1100-- Step 519: View results
1101
1102-- Step 520: View results
1103
1104-- Step 521: View results
1105
1106-- Step 522: View results
1107
1108-- Step 523: View results
1109
1110-- Step 524: View results
1111
1112-- Step 525: View results
1113
1114-- Step 526: View results
1115
1116-- Step 527: View results
1117
1118-- Step 528: View results
1119
1120-- Step 529: View results
1121
1122-- Step 530: View results
1123
1124-- Step 531: View results
1125
1126-- Step 532: View results
1127
1128-- Step 533: View results
1129
1130-- Step 534: View results
1131
1132-- Step 535: View results
1133
1134-- Step 536: View results
1135
1136-- Step 537: View results
1137
1138-- Step 538: View results
1139
1140-- Step 539: View results
1141
1142-- Step 540: View results
1143
1144-- Step 541: View results
1145
1146-- Step 542: View results
1147
1148-- Step 543: View results
1149
1150-- Step 544: View results
1151
1152-- Step 545: View results
1153
1154-- Step 546: View results
1155
1156-- Step 547: View results
1157
1158-- Step 548: View results
1159
1160-- Step 549: View results
1161
1162-- Step 550: View results
1163
1164-- Step 551: View results
1165
1166-- Step 552: View results
1167
1168-- Step 553: View results
1169
1170-- Step 554: View results
1171
1172-- Step 555: View results
1173
1174-- Step 556: View results
1175
1176-- Step 557: View results
1177
1178-- Step 558: View results
1179
1180-- Step 559: View results
1181
1182-- Step 560: View results
1183
1184-- Step 561: View results
1185
1186-- Step 562: View results
1187
1188-- Step 563: View results
1189
1190-- Step 564: View results
1191
1192-- Step 565: View results
1193
1194-- Step 566: View results
1195
1196-- Step 567: View results
1197
1198-- Step 568: View results
1199
1200-- Step 569: View results
1201
1202-- Step 570: View results
1203
1204-- Step 571: View results
1205
1206-- Step 572: View results
1207
1208-- Step 573: View results
1209
1210-- Step 574: View results
1211
1212-- Step 575: View results
1213
1214-- Step 576: View results
1215
1216-- Step 577: View results
1217
1218-- Step 578: View results
1219
1220-- Step 579: View results
1221
1222-- Step 580: View results
1223
1224-- Step 581: View results
1225
1226-- Step 582: View results
1227
1228-- Step 583: View results
1229
1230-- Step 584: View results
1231
1232-- Step 585: View results
1233
1234-- Step 586: View results
1235
1236-- Step 587: View results
1237
1238-- Step 588: View results
1239
1240-- Step 589: View results
1241
1242-- Step 590: View results
1243
1244-- Step 591: View results
1245
1246-- Step 592: View results
1247
1248-- Step 593: View results
1249
1250-- Step 594: View results
1251
1252-- Step 595: View results
1253
1254-- Step 596: View results
1255
1256-- Step 597: View results
1257
1258-- Step 598: View results
1259
1260-- Step 599: View results
1261
1262-- Step 600: View results
1263
1264-- Step 601: View results
1265
1266-- Step 602: View results
1267
1268-- Step 603: View results
1269
1270-- Step 604: View results
1271
1272-- Step 605: View results
1273
1274-- Step 606: View results
1275
1276-- Step 607: View results
1277
1278-- Step 608: View results
1279
1280-- Step 609: View results
1281
1282-- Step 610: View results
1283
1284-- Step 611: View results
1285
1286-- Step 
```

```
'Jane', 'HR', 45000.00), (103, 'Mark', 'Sales', 60000.00); INSERT INTO accounts VALUES (1, 5000.00), (2, 3000.00);
```

```
-- Step 3: Simulate ProcessMonthlyInterest UPDATE savings_accounts SET balance = balance + (balance * 0.01);
```

```
-- Step 4: Simulate UpdateEmployeeBonus DECLARE @dept VARCHAR(50) = 'Sales';  
DECLARE @bonus_percent DECIMAL(5,2) = 10;
```

```
UPDATE employees SET salary = salary + (salary * @bonus_percent / 100) WHERE department = @dept;
```

```
-- Step 5: Simulate TransferFunds DECLARE @from_acct INT = 1; DECLARE @to_acct INT = 2; DECLARE @amount DECIMAL(10,2) = 1000.00;
```

```
DECLARE @from_balance DECIMAL(10,2); SELECT @from_balance = balance FROM accounts WHERE account_id = @from_acct;
```

```
IF @from_balance >= @amount BEGIN UPDATE accounts SET balance = balance - @amount WHERE account_id = @from_acct; UPDATE accounts SET balance = balance + @amount WHERE account_id = @to_acct; END;
```

```
-- Step 6: Show output SELECT * FROM savings_accounts; SELECT * FROM employees;  
SELECT * FROM accounts;
```

OUTPUT SCREENSHOT:

```

1 -- create
2
3 -- CUSTOMER table
4 CREATE TABLE savings_accounts (
5   account_id INT PRIMARY KEY,
6   customer_name VARCHAR(50),
7   balance DECIMAL(10,2)
8 );
9
10
11 CREATE TABLE employees (
12   employee_id INT PRIMARY KEY,
13   name VARCHAR(50),
14   department VARCHAR(50),
15   salary DECIMAL(10,2)
16 );
17
18 CREATE TABLE accounts (
19   account_id INT PRIMARY KEY,
20   balance DECIMAL(10,2)
21 );
22
23 Step 2: Insert test data
24 INSERT INTO savings_accounts VALUES (1, 'Alice', 10000.00), (2, 'Bob', 8000.00);
25 INSERT INTO employees VALUES (101, 'John', 'Sales', 50000.00), (102, 'Jane', 'HR', 45000.00), (103, 'Mark', 'Sales');
26 INSERT INTO accounts VALUES (1, 50000.00), (2, 30000.00);
27
28 -- Step 3: Simulate ProcessMonthlyInterest
29 UPDATE savings_accounts
30 SET balance = balance + (balance * 0.01);
31
32 -- Step 4: Simulate UpdateEmployeeBonus
33 DECLARE @dept VARCHAR(50) = 'Sales';
34 DECLARE @bonus_percent DECIMAL(5,2) = 10;
35
36 UPDATE employees
37 SET salary = salary + (salary * @bonus_percent / 100)
38 WHERE department = @dept;
39

```

4.FUNCTION:

```
DECLARE @dob DATE = '2000-06-01';
```

```
SELECT DATEDIFF(YEAR, @dob, GETDATE()) - CASE WHEN MONTH(@dob) >
MONTH(GETDATE()) OR (MONTH(@dob) = MONTH(GETDATE()) AND DAY(@dob) >
DAY(GETDATE())) THEN 1 ELSE 0 END AS Age; DECLARE @loan_amount DECIMAL(10,2) =
100000, @annual_interest_rate DECIMAL(5,2) = 12, @years INT = 5;
```

```
DECLARE @monthly_rate DECIMAL(10,6) = @annual_interest_rate / 12 / 100, @months
INT = @years * 12, @emi DECIMAL(10,2);
```

```
IF @monthly_rate = 0 SET @emi = @loan_amount / @months; ELSE SET @emi =
(@loan_amount * @monthly_rate * POWER(1 + @monthly_rate, @months)) / (POWER(1 +
@monthly_rate, @months) - 1);
```

```
SELECT @emi AS MonthlyInstallment; CREATE TABLE accounts ( account_id INT PRIMARY
KEY, balance DECIMAL(10,2) );
```

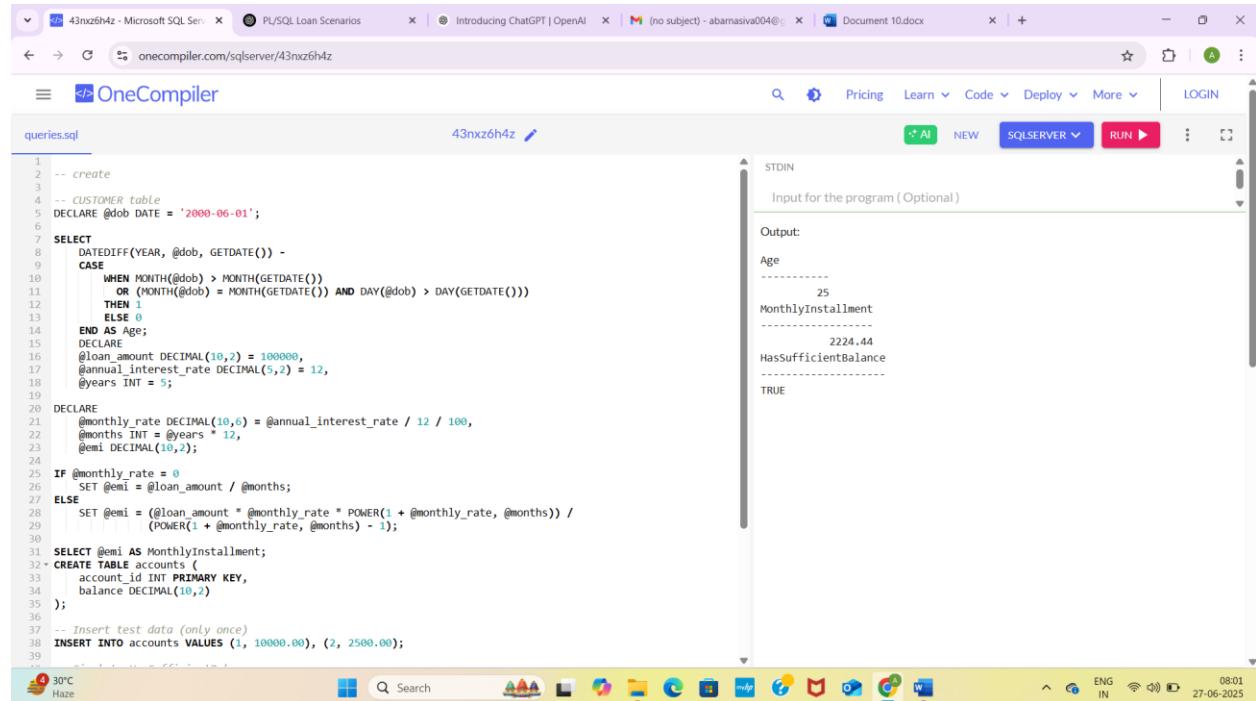
```
-- Insert test data (only once) INSERT INTO accounts VALUES (1, 10000.00), (2, 2500.00);
```

```
-- Simulate HasSufficientBalance DECLARE @acct_id INT = 1, @amount DECIMAL(10,2) =
5000, @balance DECIMAL(10,2);
```

```
SELECT @balance = balance FROM accounts WHERE account_id = @acct_id;
```

```
IF @balance IS NOT NULL AND @balance >= @amount SELECT 'TRUE' AS  
HasSufficientBalance; ELSE SELECT 'FALSE' AS HasSufficientBalance;
```

OUTPUT SCREENSHOT:



```
1 -- create
2 -- CUSTOMER table
3 DECLARE @dob DATE = '2000-06-01';
4
5 SELECT
6     DATEDIFF(YEAR, @dob, GETDATE()) -
7     CASE
8         WHEN MONTH(@dob) > MONTH(GETDATE())
9             OR (MONTH(@dob) = MONTH(GETDATE()) AND DAY(@dob) > DAY(GETDATE()))
10            THEN 1
11            ELSE 0
12        END AS Age;
13
14 DECLARE
15     @loan_amount DECIMAL(10,2) = 100000,
16     @annual_interest_rate DECIMAL(5,2) = 12,
17     @years INT = 5;
18
19 DECLARE
20     @monthly_rate DECIMAL(10,6) = @annual_interest_rate / 12 / 100,
21     @months INT = @years * 12,
22     @emi DECIMAL(10,2),
23
24 IF @monthly_rate = 0
25     SET @emi = @loan_amount / @months;
26 ELSE
27     SET @emi = (@loan_amount * @monthly_rate * POWER(1 + @monthly_rate, @months)) /
28         (POWER(1 + @monthly_rate, @months) - 1);
29
30
31 SELECT @emi AS MonthlyInstallment;
32 CREATE TABLE accounts (
33     account_id INT PRIMARY KEY,
34     balance DECIMAL(10,2)
35 );
36
37 -- Insert test data (only once)
38 INSERT INTO accounts VALUES (1, 10000.00), (2, 2500.00);
39
```

6.CURSOR

```
CREATE TABLE Customers ( customer_id INT PRIMARY KEY, name VARCHAR(50), balance  
DECIMAL(10,2) );
```

```
-- Insert sample data INSERT INTO Customers VALUES (1, 'Alice', 5000); INSERT INTO  
Customers VALUES (2, 'Bob', 6000); INSERT INTO Customers VALUES (3, 'Charlie', 7000);
```

```
-- Cursor to apply annual fee DECLARE @cust_id INT, @cust_balance DECIMAL(10,2);
```

```
DECLARE fee_cursor CURSOR FOR SELECT customer_id, balance FROM Customers;
```

```
OPEN fee_cursor; FETCH NEXT FROM fee_cursor INTO @cust_id, @cust_balance;
```

```
WHILE @@FETCH_STATUS = 0 BEGIN -- Deduct annual fee of ₹100 UPDATE Customers  
SET balance = balance - 100 WHERE customer_id = @cust_id;
```

```

PRINT 'Annual fee deducted for Customer ID: ' + CAST(@cust_id AS
VARCHAR);

FETCH NEXT FROM fee_cursor INTO @cust_id, @cust_balance;

END

CLOSE fee_cursor; DEALLOCATE fee_cursor;

-- View updated balances SELECT * FROM Customers;

-- Create Transactions table CREATE TABLE Transactions ( txn_id INT PRIMARY KEY,
customer_id INT, txn_date DATE, amount DECIMAL(10,2), txn_type VARCHAR(10) );

-- Insert sample transactions (current month) INSERT INTO Transactions VALUES (101, 1,
GETDATE(), 2000, 'deposit'); INSERT INTO Transactions VALUES (102, 2, GETDATE(), 1000,
'withdrawal'); INSERT INTO Transactions VALUES (103, 3, '2023-12-15', 3000, 'deposit'); -- old transaction

-- Cursor to print this month's transactions DECLARE @txn_customer_id INT,
@txn_amount DECIMAL(10,2), @txn_type VARCHAR(10), @txn_date DATE;

DECLARE monthly_cursor CURSOR FOR SELECT customer_id, txn_date, amount, txn_type
FROM Transactions WHERE MONTH(txn_date) = MONTH(GETDATE()) AND YEAR(txn_date)
= YEAR(GETDATE());

OPEN monthly_cursor;

FETCH NEXT FROM monthly_cursor INTO @txn_customer_id, @txn_date, @txn_amount,
@txn_type;

WHILE @@FETCH_STATUS = 0 BEGIN -- Use SELECT instead of PRINT for reliable output in
OneCompiler SELECT 'Customer ' + CAST(@txn_customer_id AS VARCHAR) + ' | Date: ' +
CAST(@txn_date AS VARCHAR) + ' | Amount: ' + CAST(@txn_amount AS VARCHAR) + ' |
Type: ' + @txn_type AS TransactionInfo;

```

```
FETCH NEXT FROM monthly_cursor INTO @txn_customer_id, @txn_date,  
@txn_amount, @txn_type;
```

END

```
CLOSE monthly_cursor; DEALLOCATE monthly_cursor;
```

```
-- To verify data also: SELECT * FROM Transactions;
```

OUTPUT SCREENSHOT:

```
queries.sql 43nxz6h4z
```

```
52 );  
53  
54 -- Insert sample transactions (current month)  
55 INSERT INTO Transactions VALUES (101, 1, GETDATE(), 2000, 'deposit');  
56 INSERT INTO Transactions VALUES (102, 2, GETDATE(), 1000, 'withdrawal');  
57 INSERT INTO Transactions VALUES (103, 3, '2023-12-15', 3000, 'deposit'); -- old transaction  
58  
59 -- cursor to print this month's transactions  
60 DECLARE @txn_customer_id INT,  
61         @txn_amount DECIMAL(10,2),  
62         @txn_type VARCHAR(10),  
63         @txn_date DATE;  
64  
65 DECLARE monthly_cursor CURSOR FOR  
66     SELECT customer_id, txn_date, amount, txn_type  
67     FROM Transactions  
68     WHERE MONTH(txn_date) = MONTH(GETDATE()) AND YEAR(txn_date) = YEAR(GETDATE());  
69  
70 OPEN monthly_cursor;  
71  
72 FETCH NEXT FROM monthly_cursor INTO @txn_customer_id, @txn_date, @txn_amount, @txn_type;  
73  
74 WHILE @@FETCH_STATUS = 0  
75 BEGIN  
76     -- Use SELECT instead of PRINT for reliable output in OneCompiler  
77     SELECT CustomerName + CAST(@txn_customer_id AS VARCHAR) +  
78             ' | Date: ' + CAST(@txn_date AS VARCHAR) +  
79             ' | Amount: ' + CAST(@txn_amount AS VARCHAR) +  
80             ' | Type: ' + @txn_type AS TransactionInfo;  
81  
82     FETCH NEXT FROM monthly_cursor INTO @txn_customer_id, @txn_date, @txn_amount, @txn_type;  
83  
84 END  
85  
86 CLOSE monthly_cursor;  
87 DEALLOCATE monthly_cursor;  
88  
89 -- To verify data also:  
90 SELECT * FROM Transactions;
```

Output:

```
Annual fee deducted for Customer ID: 1  
Annual fee deducted for Customer ID: 2  
Annual fee deducted for Customer ID: 3  
customer_id name balance  
-----  
1 Alice 4900.00  
2 Bob 5900.00  
3 Charlie 6900.00  
TransactionInfo  
-----  
Customer 1 | Date: 2025-06-27 | Amount: 2000.00 | Type: deposit  
TransactionInfo  
-----  
Customer 2 | Date: 2025-06-27 | Amount: 1000.00 | Type: withdrawal  
txnid customer_id txn_date amount txn_type  
-----  
101 1 2025-06-27 2000.00 deposit  
102 2 2025-06-27 1000.00 withdrawal  
103 3 2023-12-15 3000.00 deposit
```

7.PACKAGES:

IF OBJECT_ID('Customers', 'U') IS NOT NULL

```
DROP TABLE Customers;
```

```
-- Create Customers table
```

```
CREATE TABLE Customers (
```

```
    CustomerID INT PRIMARY KEY,
```

```
    Name VARCHAR(100),
```

```
    DOB DATE,  
    Balance DECIMAL(10, 2),  
    LastModified DATETIME  
);
```

```
-- Create stored procedure to add a new customer  
IF OBJECT_ID('AddCustomer', 'P') IS NOT NULL  
    DROP PROCEDURE AddCustomer;  
GO
```

```
CREATE PROCEDURE AddCustomer
```

```
    @CustomerID INT,  
    @Name VARCHAR(100),  
    @DOB DATE,  
    @Balance DECIMAL(10,2)
```

```
AS
```

```
BEGIN
```

```
    INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)  
    VALUES (@CustomerID, @Name, @DOB, @Balance, GETDATE());  
END;  
GO
```

```
-- Create stored procedure to update customer details
```

```
IF OBJECT_ID('UpdateCustomer', 'P') IS NOT NULL  
    DROP PROCEDURE UpdateCustomer;  
GO
```

```
CREATE PROCEDURE UpdateCustomer
    @CustomerID INT,
    @Name VARCHAR(100),
    @Balance DECIMAL(10,2)
AS
BEGIN
    UPDATE Customers
    SET Name = @Name,
        Balance = @Balance,
        LastModified = GETDATE()
    WHERE CustomerID = @CustomerID;
END;
GO

-- Create function to get customer balance
IF OBJECT_ID('GetBalance', 'FN') IS NOT NULL
    DROP FUNCTION GetBalance;
GO

CREATE FUNCTION GetBalance (@CustomerID INT)
RETURNS DECIMAL(10,2)
AS
BEGIN
    DECLARE @bal DECIMAL(10,2);
    SELECT @bal = Balance FROM Customers WHERE CustomerID = @CustomerID;
    RETURN @bal;
END;
```

```
    RETURN @bal;
```

```
END;
```

```
GO
```

```
--  Test the procedures and function
```

```
-- Add a new customer
```

```
EXEC AddCustomer 1, 'John Doe', '1990-01-01', 5000;
```

```
-- Update customer info
```

```
EXEC UpdateCustomer 1, 'John D.', 5500;
```

```
-- Get balance using the function
```

```
SELECT dbo.GetBalance(1) AS CustomerBalance;
```

```
-- View the full customer record
```

```
SELECT * FROM Customers;
```

```
OUTPUT SCREENSHOT:
```

The screenshot shows the OneCompiler interface with a SQL script named 'queries.sql' containing several database operations. The output pane displays the results of a query that retrieves customer balance information.

```

1 -- create
2 IF OBJECT_ID('Customers', 'U') IS NOT NULL
3     DROP TABLE Customers;
4
5 -- Create Customers table
6 CREATE TABLE Customers (
7     CustomerID INT PRIMARY KEY,
8     Name VARCHAR(100),
9     DOB DATE,
10    Balance DECIMAL(10, 2),
11    LastModified DATETIME
12 );
13
14 -- Create stored procedure to add a new customer
15 IF OBJECT_ID('AddCustomer', 'P') IS NOT NULL
16     DROP PROCEDURE AddCustomer;
17 GO
18
19 CREATE PROCEDURE AddCustomer
20     @CustomerID INT,
21     @Name VARCHAR(100),
22     @DOB DATE,
23     @Balance DECIMAL(10,2)
24 AS
25 BEGIN
26     INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)
27     VALUES (@CustomerID, @Name, @DOB, @Balance, GETDATE());
28 END;
29 GO
30
31 -- Create stored procedure to update customer details
32 IF OBJECT_ID('UpdateCustomer', 'P') IS NOT NULL
33     DROP PROCEDURE UpdateCustomer;
34 GO
35
36 CREATE PROCEDURE UpdateCustomer
37     @CustomerID INT,
38     @Name VARCHAR(100),
39     ...

```

Output:

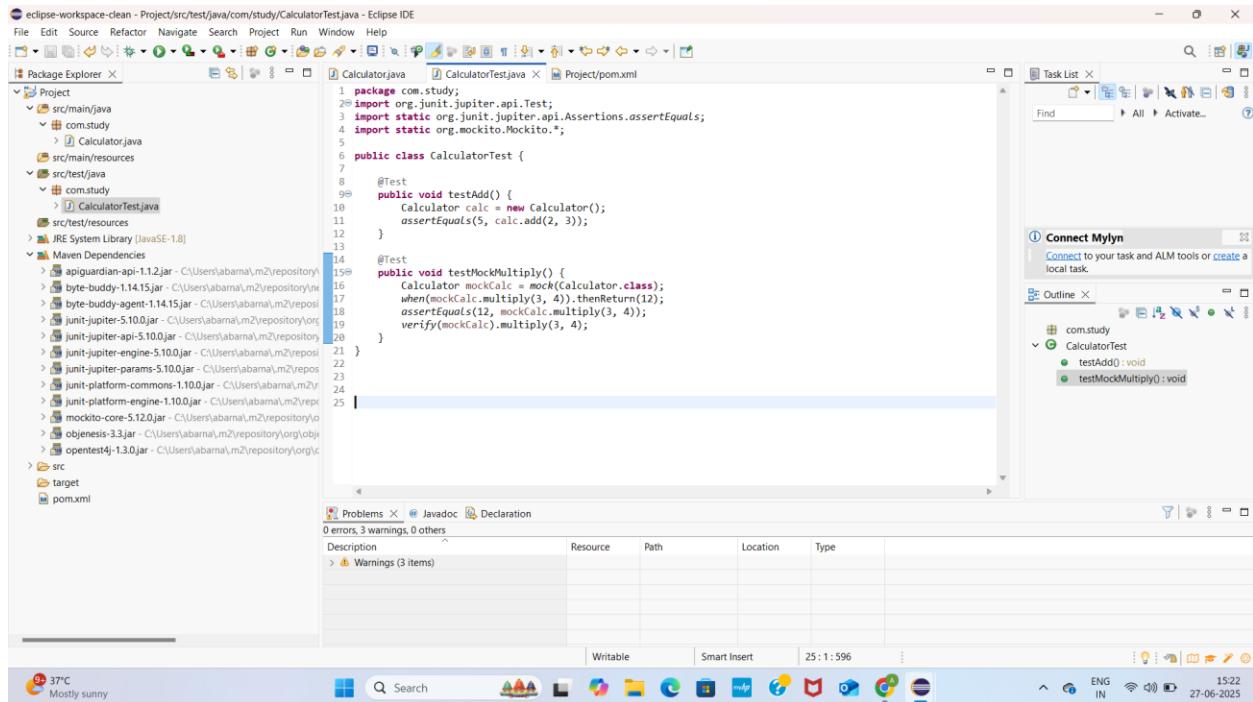
CustomerBalance	
5500.00	
CustomerID	Name
1	John D.

This screenshot is similar to the first one, but it highlights specific columns in the output table. The 'DOB', 'Balance', and 'LastModified' columns are explicitly labeled at the top of the table.

DOB	Balance	LastModified
1990-01-01	5500.00	2025-06-27 03:54:10.000

1. Setting up Junit(example test)

Test class is created and new java project in eclipse



Pom.xml

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.study</groupId>
  <artifactId>Project</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>

  <dependencies>
    <!-- JUnit 5 -->
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter</artifactId>
      <version>5.10.0</version>
      <scope>test</scope>
    </dependency>
  
```

```

<!-- ✅ Mockito -->
<dependency>
<groupId>org.mockito</groupId>
<artifactId>mockito-core</artifactId>
<version>5.12.0</version>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<!-- Plugin to run JUnit 5 tests -->
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>3.1.2</version>
<configuration>
<useModulePath>false</useModulePath>
</configuration>
</plugin>
</plugins>
</build>

</project>

```

By using maven I have added it you could view in screenshot

2.Writing Basic Junit test

Calculator.java

```
package com.study;
```

```
public class Calculator {  
    public int add(int a, int b) {  
        return a + b;  
    }  
  
    public int multiply(int a, int b) {  
        return a * b;  
    }  
}
```

Calculatortest.java

```
package com.study;  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.assertEquals;  
import static org.mockito.Mockito.*;
```

```
public class CalculatorTest {
```

```
@Test  
public void testAdd() {  
    Calculator calc = new Calculator();  
    assertEquals(5, calc.add(2, 3));  
}
```

```
@Test  
public void testMockMultiply() {  
    Calculator mockCalc = mock(Calculator.class);  
    when(mockCalc.multiply(3, 4)).thenReturn(12);  
    assertEquals(12, mockCalc.multiply(3, 4));  
    verify(mockCalc).multiply(3, 4);  
}
```

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
    http://maven.apache.org/xsd/maven-4.0.0.xsd">  
    <modelVersion>4.0.0</modelVersion>  
  
    <groupId>com.study</groupId>  
    <artifactId>Project</artifactId>  
    <version>1.0-SNAPSHOT</version>  
  
    <properties>
```

```

<maven.compiler.source>1.8</maven.compiler.source>
<maven.compiler.target>1.8</maven.compiler.target>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>

<dependencies>
  <!--  JUnit 5 -->
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter</artifactId>
    <version>5.10.0</version>
    <scope>test</scope>
  </dependency>

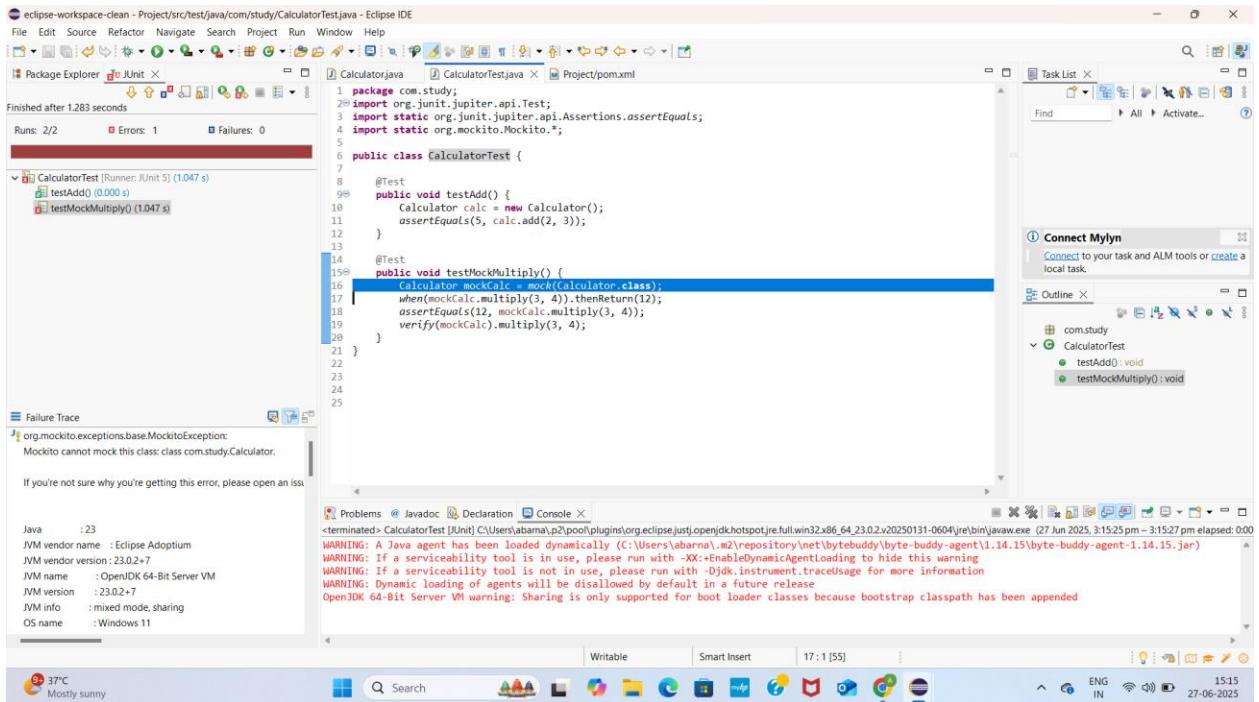
  <!--  Mockito -->
  <dependency>
    <groupId>org.mockito</groupId>
    <artifactId>mockito-core</artifactId>
    <version>5.12.0</version>
    <scope>test</scope>
  </dependency>
</dependencies>

<build>
  <plugins>
    <!-- Plugin to run JUnit 5 tests -->
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>3.1.2</version>
      <configuration>
        <useModulePath>false</useModulePath>
      </configuration>
    </plugin>
  </plugins>
</build>

</project>

```

OUTPUT SCREENSHOT:



3.Assertion in junit

AssertionsTest.java

```

package com.study;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class AssertionsTest {

    @Test
    public void testAssertions() {
        // Assert equals
        assertEquals(5, 2 + 3);

        // Assert true
        assertTrue(5 > 3);

        // Assert false
        assertFalse(5 < 3);

        // Assert null
        assertNull(null);

        // Assert not null
        assertNotNull(new Object());
    }
}

```

```
}
```

```
Pom.xml
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<groupId>com.study</groupId>
<artifactId>Project</artifactId>
<version>1.0-SNAPSHOT</version>

<properties>
<maven.compiler.source>1.8</maven.compiler.source>
<maven.compiler.target>1.8</maven.compiler.target>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>

<dependencies>
<!-- JUnit 5 -->
<dependency>
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter</artifactId>
<version>5.10.0</version>
<scope>test</scope>
</dependency>

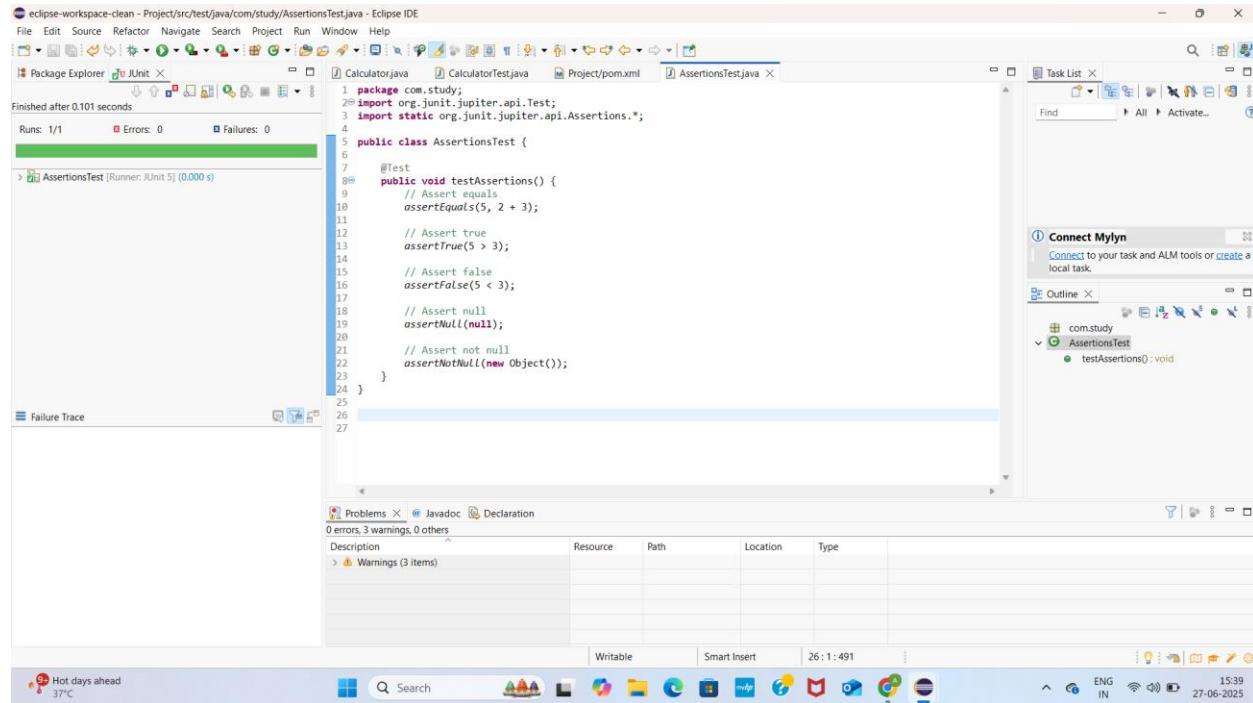
<!-- Mockito -->
<dependency>
<groupId>org.mockito</groupId>
<artifactId>mockito-core</artifactId>
<version>5.12.0</version>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<!-- Plugin to run JUnit 5 tests -->
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>3.1.2</version>
<configuration>
<useModulePath>false</useModulePath>
```

```
</configuration>
</plugin>
</plugins>
</build>
```

```
</project>
```

OUTPUTSCREENSHOT:



4. Arrange Act Assert

CalculatorTest.java

```
package com.study;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.assertEquals;
import static org.mockito.Mockito.*;

public class CalculatorTest {

    @Test
    public void testAdd() {
        Calculator calc = new Calculator();
        assertEquals(5, calc.add(2, 3));
    }
}
```

```
@Test
public void testMockMultiply() {
Calculator mockCalc = mock(Calculator.class);
when(mockCalc.subtract(3, 4)).thenReturn(12);
assertEquals(12, mockCalc.subtract(3, 4));
verify(mockCalc).subtract(3, 4);
}
}

CalculatorTestLifecycle.java

package com.study;
import org.junit.jupiter.api.*;

import static org.junit.jupiter.api.Assertions.*;

public class CalculatorLifecycleTest {

private Calculator calc; // Declare calc globally (class-level)

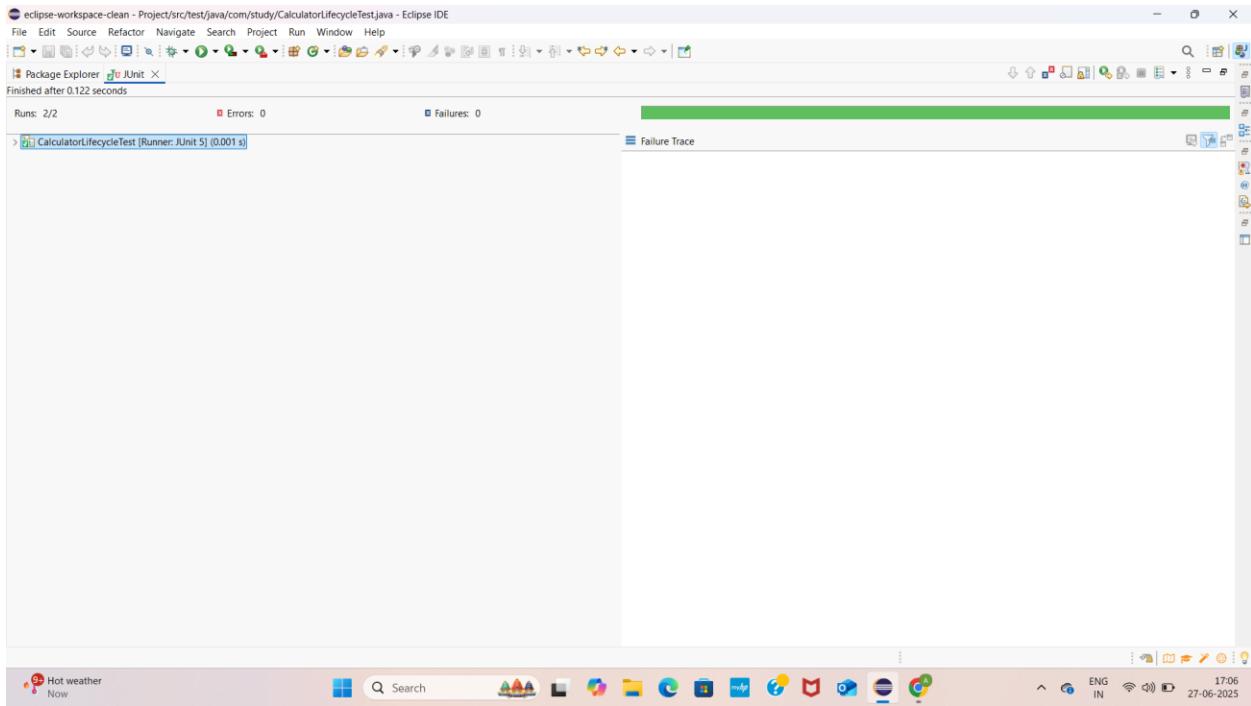
@BeforeEach
public void setUp() {
calc = new Calculator(); // Initialize it before each test
}

@AfterEach
public void tearDown() {
calc = null; // Clean up
}

@Test
public void testAdd() {
int result = calc.add(2, 3);
assertEquals(5, result);
}

@Test
public void testSubtract() {
int result = calc.subtract(10, 4);
assertEquals(6, result);
}
}
```

OUTPUTSCREENSHOT



ADVANCED JUNIT TESTING EXERCISES

1. PARAMETERIZED TEST

EvenChecker.java

```
package com.study;

public class EvenChecker {
    public boolean isEven(int number) {
        return number % 2 == 0;
    }
}
```

EvenCheckerTest.java

```
package com.study;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.ValueSource;

import static org.junit.jupiter.api.Assertions.*;

public class EvenCheckerTest {

    private final EvenChecker checker = new EvenChecker();

    @ParameterizedTest
    @ValueSource(ints = {2, 4, 6, 10, 100})
}
```

```
public void testEvenNumbers(int number) {  
    assertTrue(checker.isEven(number), number + " should be even");  
}  
}
```

```
@ParameterizedTest  
@ValueSource(ints = {1, 3, 5, 9, 101})  
public void testOddNumbers(int number) {  
    assertFalse(checker.isEven(number), number + " should be odd");  
}  
}  
}
```

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
        http://maven.apache.org/xsd/maven-4.0.0.xsd">  
    <modelVersion>4.0.0</modelVersion>  
  
    <groupId>com.study</groupId>  
    <artifactId>Project</artifactId>  
    <version>1.0-SNAPSHOT</version>  
  
    <properties>  
        <maven.compiler.source>1.8</maven.compiler.source>  
        <maven.compiler.target>1.8</maven.compiler.target>  
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
    </properties>  
  
    <dependencies>  
        <!-- JUnit 5 -->  
        <dependency>  
            <groupId>org.junit.jupiter</groupId>  
            <artifactId>junit-jupiter</artifactId>  
            <version>5.10.0</version>  
            <scope>test</scope>  
        </dependency>  
  
        <!-- Mockito -->  
        <dependency>  
            <groupId>org.mockito</groupId>  
            <artifactId>mockito-core</artifactId>  
            <version>5.12.0</version>  
            <scope>test</scope>  
        </dependency>  
    </dependencies>
```

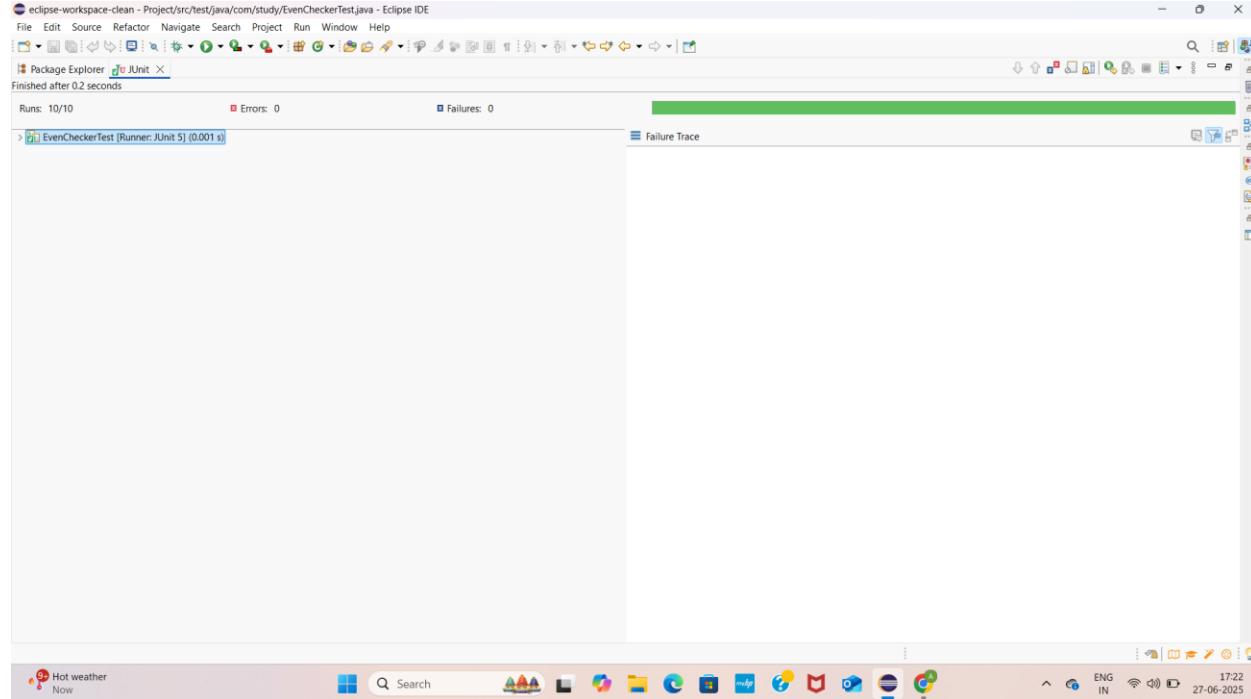
```

<build>
<plugins>
    <!-- Plugin to run JUnit 5 tests -->
    <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-surefire-plugin</artifactId>
        <version>3.1.2</version>
        <configuration>
            <useModulePath>false</useModulePath>
        </configuration>
    </plugin>
</plugins>
</build>

</project>

```

OUTPUTSCREENSHOT



2.Test Suites and categories

AllTests.java

```

package com.study;

import org.junit.jupiter.api.Assertions;

public class AllTests {

```

```
public static void main(String[] args) {
CalculatorLifecycleTest calculatorTest = new CalculatorLifecycleTest();
EvenCheckerTest evenCheckerTest = new EvenCheckerTest();

// Manually call each test (setup not called automatically)
try {
calculatorTest.setUp();
calculatorTest.testAdd();
calculatorTest.testSubtract();
calculatorTest.tearDown();

evenCheckerTest.testEvenNumbers(2);
evenCheckerTest.testEvenNumbers(4);
evenCheckerTest.testEvenNumbers(6);
evenCheckerTest.testEvenNumbers(10);
evenCheckerTest.testEvenNumbers(100);

evenCheckerTest.testOddNumbers(1);
evenCheckerTest.testOddNumbers(3);
evenCheckerTest.testOddNumbers(5);
evenCheckerTest.testOddNumbers(9);
evenCheckerTest.testOddNumbers(101);

System.out.println("☑ All tests passed manually!");

} catch (AssertionError | Exception e) {
System.out.println("✗ Test failed: " + e.getMessage();
}
}
}

EvenChecker.java
```

```
package com.study;

public class EvenChecker {
public boolean isEven(int number) {
return number % 2 == 0;
}
}
```

CalculatorLifecycleTest.java

```
package com.study;
import org.junit.jupiter.api.*;

import static org.junit.jupiter.api.Assertions.*;
@Tag("math")
```

```

public class CalculatorLifecycleTest {

    private Calculator calc; // Declare calc globally (class-level)

    @BeforeEach
    public void setUp() {
        calc = new Calculator(); // Initialize it before each test
    }

    @AfterEach
    public void tearDown() {
        calc = null; // Clean up
    }

    @Test
    public void testAdd() {
        int result = calc.add(2, 3);
        assertEquals(5, result, "Addition failed");
    }

    @Test
    public void testSubtract() {
        int result = calc.subtract(10, 4);
        assertEquals(6, result, "Subtraction failed");
    }
}

```

Pom.xml

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.study</groupId>
    <artifactId>Project</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

```

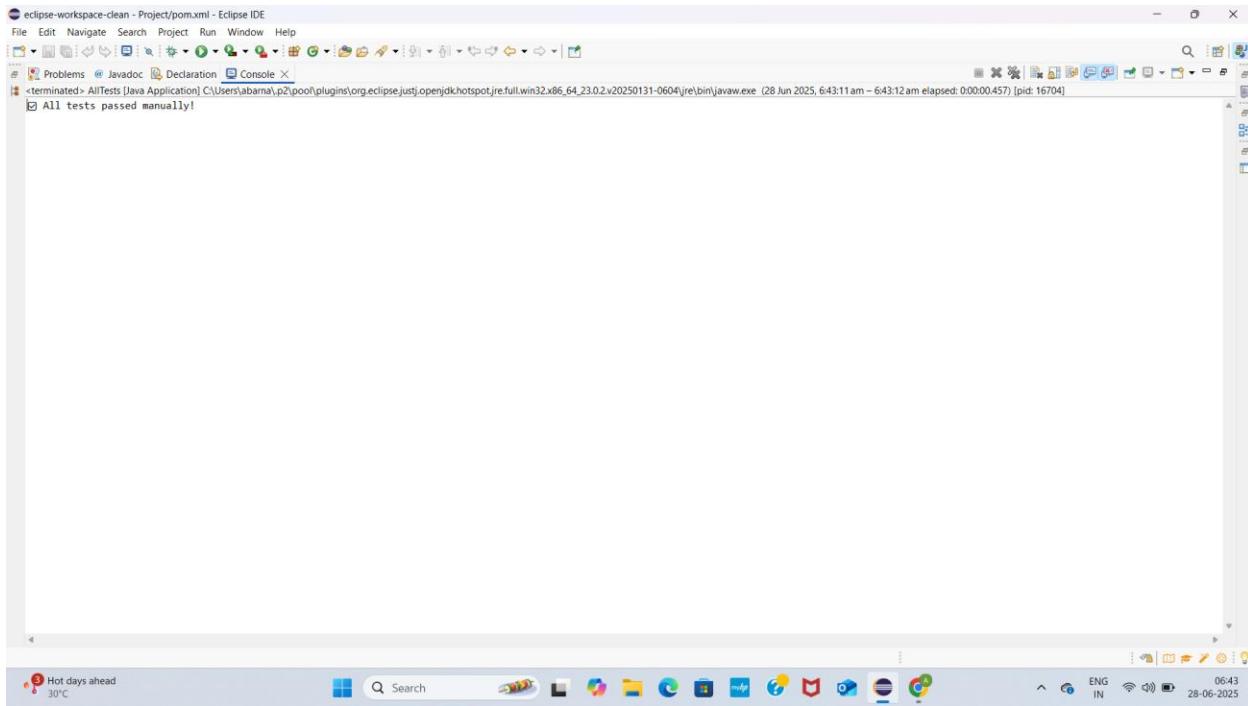
```
<dependencies>
  <!-- JUnit 5 -->
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter</artifactId>
    <version>5.10.0</version>
    <scope>test</scope>
  </dependency>

  <!-- Mockito -->
  <dependency>
    <groupId>org.mockito</groupId>
    <artifactId>mockito-core</artifactId>
    <version>5.12.0</version>
    <scope>test</scope>
  </dependency>
</dependencies>

<build>
  <plugins>
    <!-- Plugin to run JUnit 5 tests -->
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>3.1.2</version>
      <configuration>
        <useModulePath>false</useModulePath>
      </configuration>
    </plugin>
  </plugins>
</build>

</project>
```

Output screenshot



3. Test Execution Order

OrderedTest.java

```
package com.study;
import org.junit.jupiter.api.*;

import org.junit.jupiter.api.MethodOrderer.OrderAnnotation;

@TestMethodOrder(OrderAnnotation.class) // Tell JUnit to use @Order
public class OrderedTests {

    @Test
    @Order(1)
    void testFirst() {
        System.out.println("Running testFirst()");
    }

    @Test
    @Order(3)
    void testThird() {
        System.out.println("Running testThird()");
    }

    @Test
    @Order(2)
    void testSecond() {
```

```
System.out.println("Running testSecond()");  
}  
}  
}
```

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
        http://maven.apache.org/xsd/maven-4.0.0.xsd">  
    <modelVersion>4.0.0</modelVersion>  
  
    <groupId>com.study</groupId>  
    <artifactId>Project</artifactId>  
    <version>1.0-SNAPSHOT</version>  
  
    <properties>  
        <maven.compiler.source>1.8</maven.compiler.source>  
        <maven.compiler.target>1.8</maven.compiler.target>  
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
    </properties>  
  
    <dependencies>  
        <!--  JUnit 5 -->  
        <dependency>  
            <groupId>org.junit.jupiter</groupId>  
            <artifactId>junit-jupiter</artifactId>  
            <version>5.10.0</version>  
            <scope>test</scope>  
        </dependency>  
  
        <!--  Mockito -->  
        <dependency>  
            <groupId>org.mockito</groupId>  
            <artifactId>mockito-core</artifactId>  
            <version>5.12.0</version>  
            <scope>test</scope>  
        </dependency>  
    </dependencies>  
  
    <build>  
        <plugins>  
            <!-- Plugin to run JUnit 5 tests -->  
            <plugin>  
                <groupId>org.apache.maven.plugins</groupId>  
                <artifactId>maven-surefire-plugin</artifactId>  
                <version>3.1.2</version>  
            </plugin>  
        </plugins>  
    </build>  
</project>
```

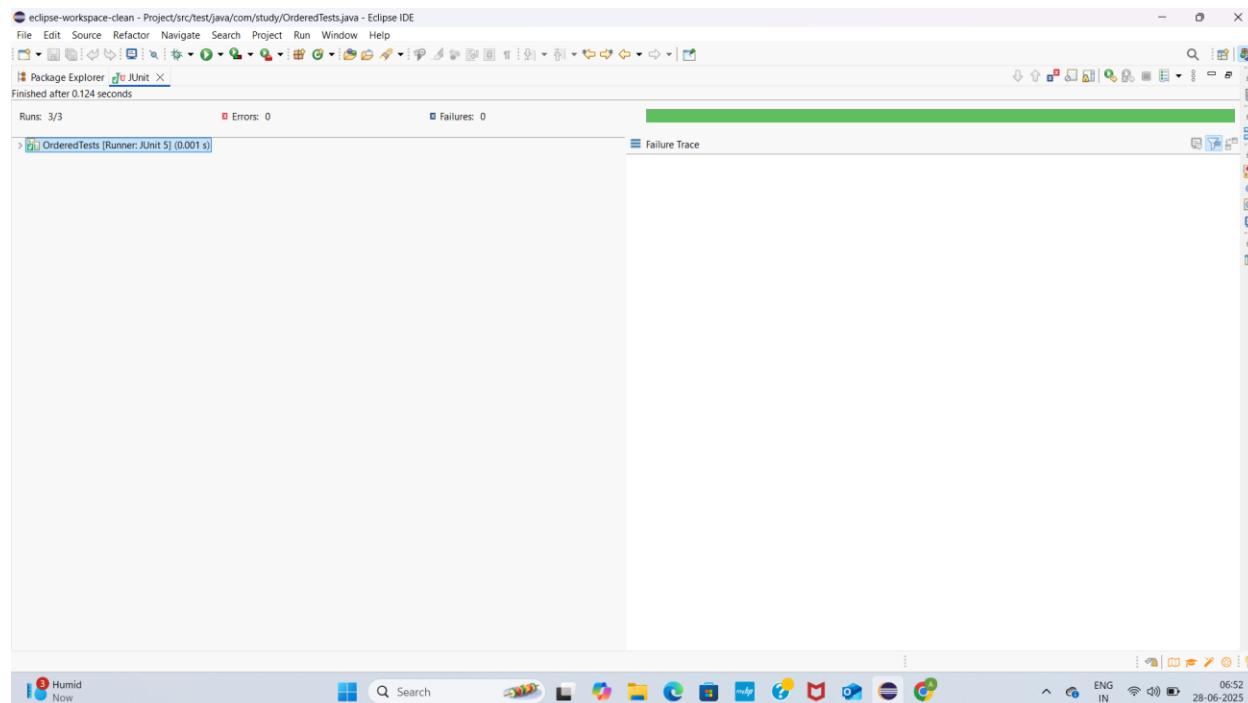
```

<configuration>
<useModulePath>false</useModulePath>
</configuration>
</plugin>
</plugins>
</build>

</project>

```

OUTPUT SCREENSHOT:



4.Exception Testing

ExceptionThrowerTest.java

```

package com.study;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class ExceptionThrowerTest {

    @Test
    void testThrowException_whenInputIsNull_shouldThrowException() {
        ExceptionThrower thrower = new ExceptionThrower();

        Exception exception = assertThrows(IllegalArgumentException.class, () -> {
            thrower.throwException(null);
        });

        assertEquals("Input cannot be null or empty", exception.getMessage());
    }
}

```

```
}
```

```
@Test
void testThrowException_whenInputIsEmpty_shouldThrowException() {
    ExceptionThrower thrower = new ExceptionThrower();

    Exception exception = assertThrows(IllegalArgumentException.class, () -> {
        thrower.throwException("");
    });
}
```

ExceptionThrower.java

```
package com.study;

public class ExceptionThrower {

    public void throwException(String input) {
        if (input == null || input.isEmpty()) {
            throw new IllegalArgumentException("Input cannot be null or empty");
        }
        // else logic here
    }
}
```

```
assertEquals("Input cannot be null or empty", exception.getMessage());
}
}
Pom.xml
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.study</groupId>
    <artifactId>Project</artifactId>
    <version>1.0-SNAPSHOT</version>

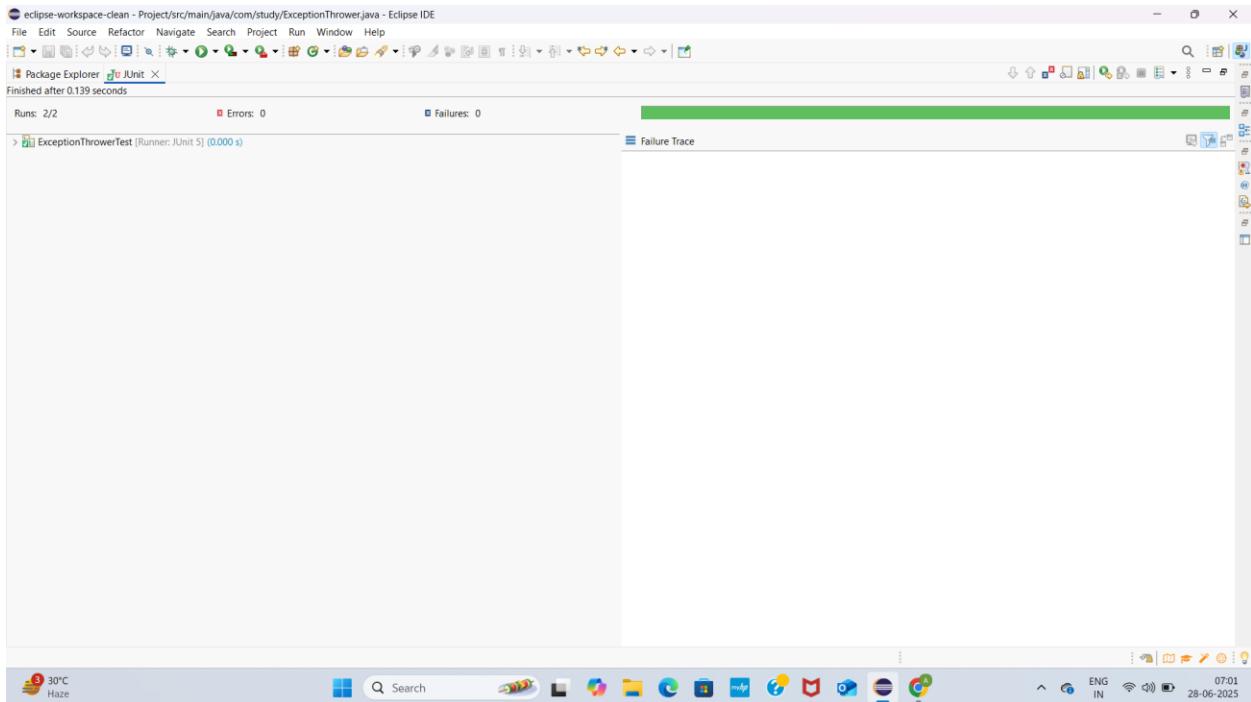
    <properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>
```

```
<dependencies>
    <!-- JUnit 5 -->
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter</artifactId>
        <version>5.10.0</version>
        <scope>test</scope>
    </dependency>

    <!-- Mockito -->
    <dependency>
        <groupId>org.mockito</groupId>
        <artifactId>mockito-core</artifactId>
        <version>5.12.0</version>
        <scope>test</scope>
    </dependency>
</dependencies>

<build>
    <plugins>
        <!-- Plugin to run JUnit 5 tests -->
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-surefire-plugin</artifactId>
            <version>3.1.2</version>
            <configuration>
                <useModulePath>false</useModulePath>
            </configuration>
        </plugin>
    </plugins>
</build>
</project>
```

OUTPUTSCREENSHOT



5. TIMEOUT AND PERFORMANCE TESTING

PerformanceTesterTest.java

```
package com.study;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.assertTimeout;
import java.time.Duration;

public class PerformanceTesterTest {

    @Test
    void testPerformTaskCompletesWithin500Milliseconds() {
        PerformanceTester tester = new PerformanceTester();

        // Assert that performTask() completes within 500 milliseconds
        assertTimeout(Duration.ofMillis(500), tester::performTask);
    }
}
```

PerformanceTester.java

```
package com.study;

public class PerformanceTester {

    public void performTask() {
        // Simulate a task that takes time (e.g., 300 milliseconds)
    }
}
```

```

try {
    Thread.sleep(300);
} catch (InterruptedException e) {
    Thread.currentThread().interrupt(); // good practice
}
}

Pom.xml

<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<groupId>com.study</groupId>
<artifactId>Project</artifactId>
<version>1.0-SNAPSHOT</version>

<properties>
<maven.compiler.source>1.8</maven.compiler.source>
<maven.compiler.target>1.8</maven.compiler.target>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>

<dependencies>
<!--  JUnit 5 -->
<dependency>
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter</artifactId>
<version>5.10.0</version>
<scope>test</scope>
</dependency>

<!--  Mockito -->
<dependency>
<groupId>org.mockito</groupId>
<artifactId>mockito-core</artifactId>
<version>5.12.0</version>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<!-- Plugin to run JUnit 5 tests -->

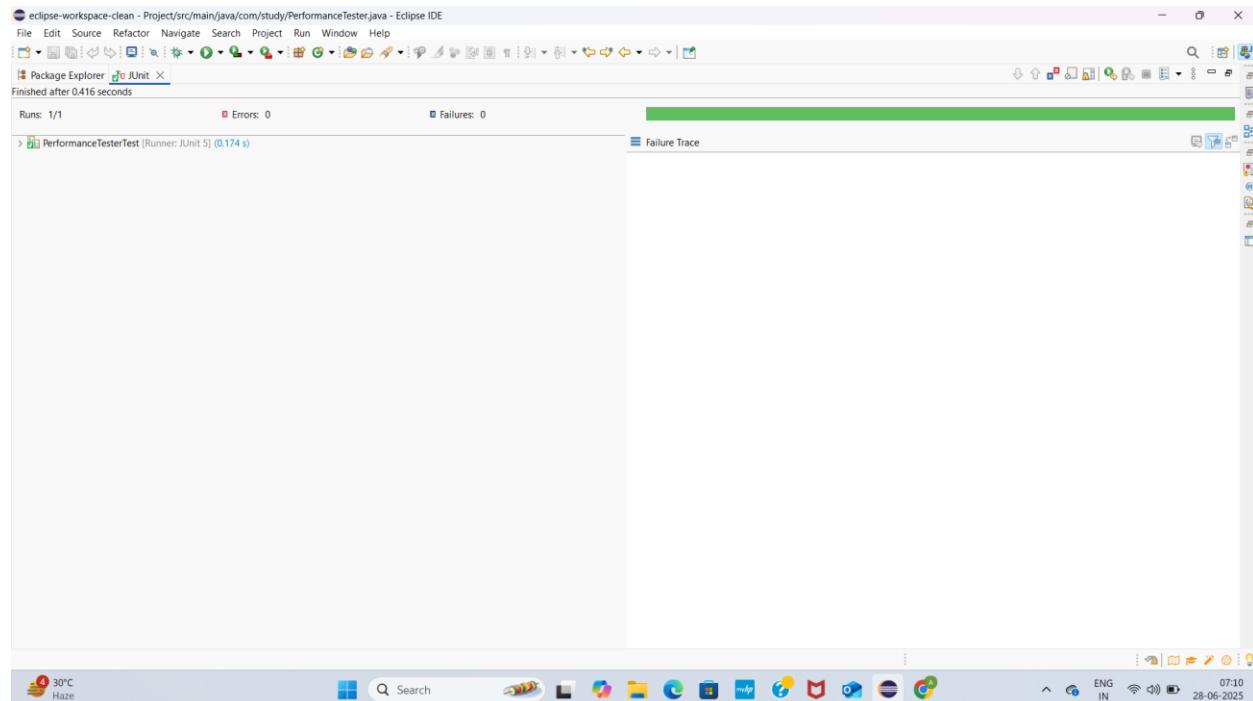
```

```

<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>3.1.2</version>
<configuration>
<useModulePath>false</useModulePath>
</configuration>
</plugin>
</plugins>
</build>
</project>

```

OUTPUT SCREENSHOT



MOCKITO HANDS ON

1. MOCKING AND STUBBING:

```

MyServiceTest.java
package com.study;
import static org.mockito.Mockito.*;
import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.api.Test;
import org.mockito.Mockito;

public class MyServiceTest {

```

```
@Test
public void testExternalApi() {
// Step 1: Create a mock object
ExternalApi mockApi = Mockito.mock(ExternalApi.class);

// Step 2: Stub the method
when(mockApi.getData()).thenReturn("Mock Data");

// Step 3: Inject mock into the service and test
MyService service = new MyService(mockApi);
String result = service.fetchData();

// Assert that mock data is returned
assertEquals("Mock Data", result);
}
}
```

MyService.java

```
package com.study;

public class MyService {

private ExternalApi externalApi;

public MyService(ExternalApi externalApi) {
this.externalApi = externalApi;
}

public String fetchData() {
return externalApi.getData(); // Calls the real or mocked API
}
}
```

ExternalApi.java

```
package com.study;
```

```
public interface ExternalApi {
String getData();
```

```
}
```

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
```

```
http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<groupId>com.study</groupId>
<artifactId>Project</artifactId>
<version>1.0-SNAPSHOT</version>

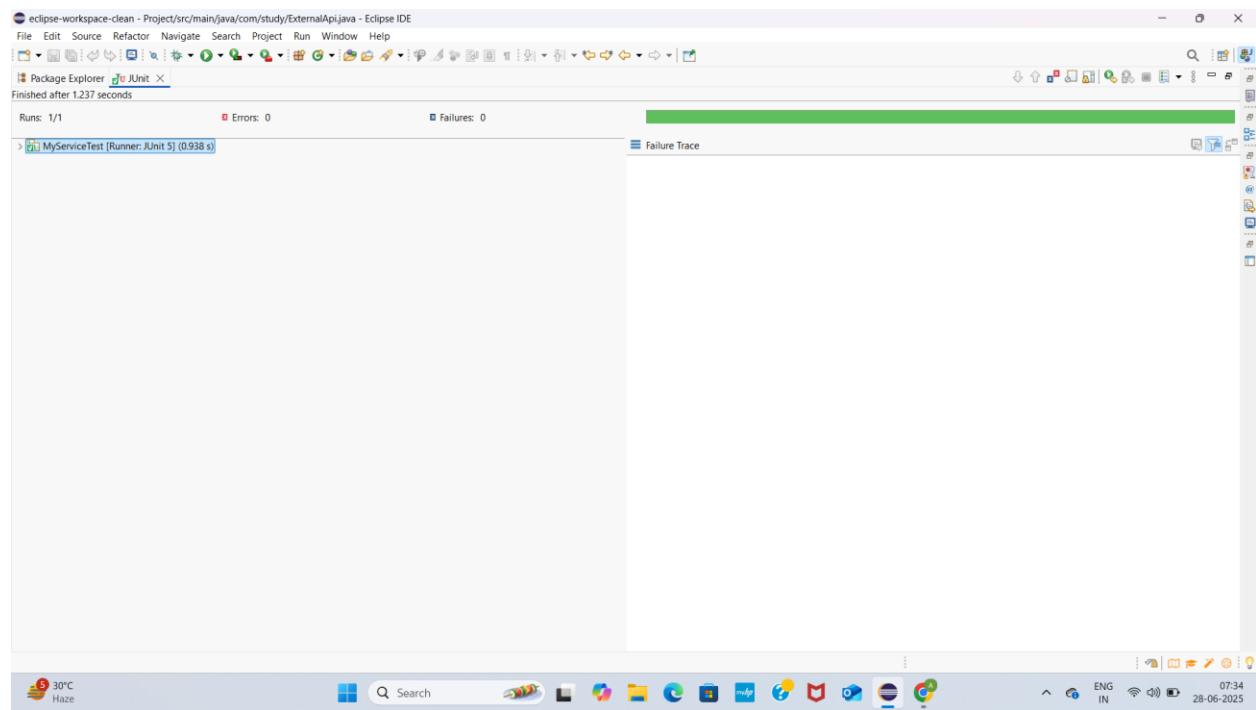
<properties>
<maven.compiler.source>1.8</maven.compiler.source>
<maven.compiler.target>1.8</maven.compiler.target>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>

<dependencies>
<!--  JUnit 5 -->
<dependency>
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter</artifactId>
<version>5.10.0</version>
<scope>test</scope>
</dependency>

<!--  Mockito -->
<dependency>
<groupId>org.mockito</groupId>
<artifactId>mockito-core</artifactId>
<version>5.12.0</version>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<!-- Plugin to run JUnit 5 tests -->
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>3.1.2</version>
<configuration>
<useModulePath>false</useModulePath>
</configuration>
</plugin>
</plugins>
</build>
</project>
```

OUTPUT SCREENSHOT



MyServiceTest.java

```
package com.study;
import static org.mockito.Mockito.*;
import org.junit.jupiter.api.Test;

import org.mockito.Mockito;
```

public class MyServiceTest {

```
@Test
public void testVerifyInteraction() {
// Step 1: Create a mock object
ExternalApi mockApi = Mockito.mock(ExternalApi.class);
MyService service = new MyService(mockApi);

service.fetchData();
verify(mockApi).getData();
}
```

MyService.java

```
package com.study;
```

```
public class MyService {  
  
    private ExternalApi externalApi;  
  
    public MyService(ExternalApi externalApi) {  
        this.externalApi = externalApi;  
    }  
  
    public String fetchData() {  
        return externalApi.getData(); // Calls the real or mocked API  
    }  
}
```

ExternalApi.java

```
package com.study;
```

```
public interface ExternalApi {  
    String getData();
```

}

Pom.xml

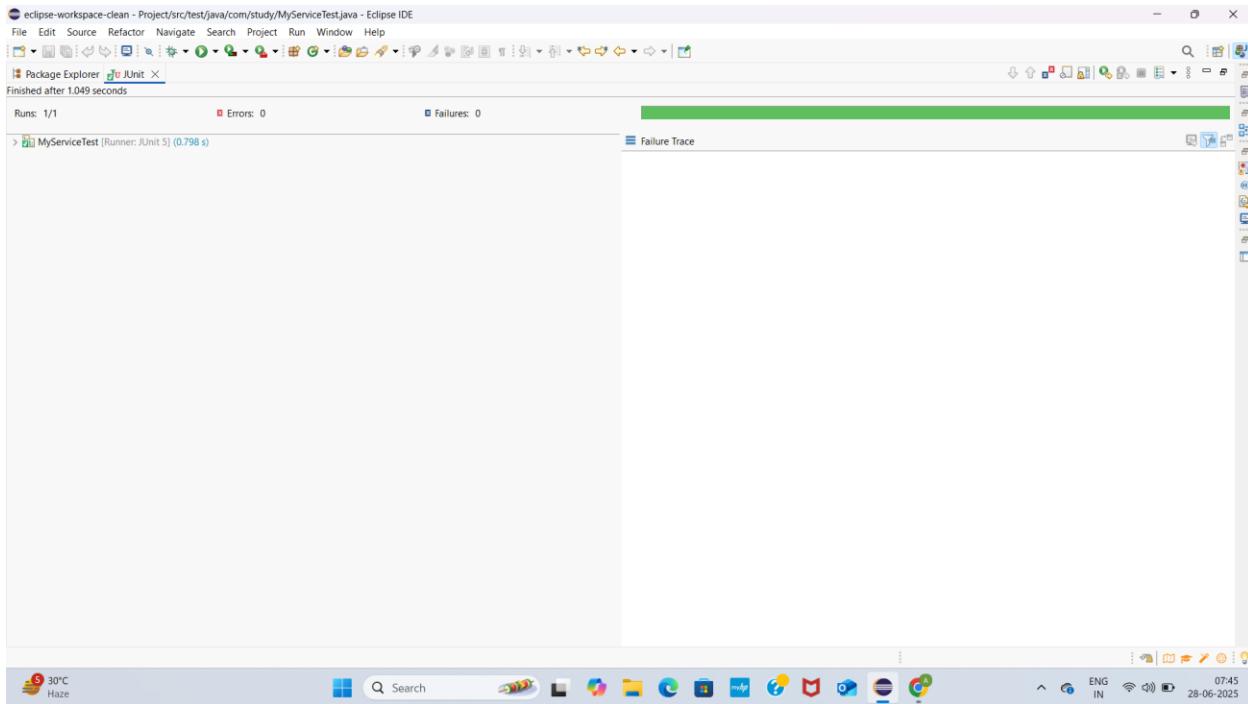
```
<project xmlns="http://maven.apache.org/POM/4.0.0"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
    http://maven.apache.org/xsd/maven-4.0.0.xsd">  
    <modelVersion>4.0.0</modelVersion>  
  
    <groupId>com.study</groupId>  
    <artifactId>Project</artifactId>  
    <version>1.0-SNAPSHOT</version>  
  
    <properties>  
        <maven.compiler.source>1.8</maven.compiler.source>  
        <maven.compiler.target>1.8</maven.compiler.target>  
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
    </properties>  
  
    <dependencies>  
        <!--  JUnit 5 -->  
        <dependency>  
            <groupId>org.junit.jupiter</groupId>  
            <artifactId>junit-jupiter</artifactId>  
            <version>5.10.0</version>  
            <scope>test</scope>
```

```
</dependency>

<!-- ☑ Mockito -->
<dependency>
<groupId>org.mockito</groupId>
<artifactId>mockito-core</artifactId>
<version>5.12.0</version>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<!-- Plugin to run JUnit 5 tests -->
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>3.1.2</version>
<configuration>
<useModulePath>false</useModulePath>
</configuration>
</plugin>
</plugins>
</build>
</project>
```

OUTPUTSCREENSHOT



3. ARGUMENT MATCHING

```
MyServiceTest.java
package com.study;
import static org.mockito.Mockito.*;
import static org.mockito.ArgumentMatchers.*;
import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

    @Test
    public void testArgumentMatching() {
        // Step 1: Create mock
        ExternalApi mockApi = Mockito.mock(ExternalApi.class);

        // Step 2: Inject mock and call method with specific argument
        MyService service = new MyService(mockApi);
        service.process("Hello World");

        // Step 3: Verify it was called with specific argument using matcher
        verify(mockApi).sendData(eq("Hello World")); //  Exact match

        // You could also match any string like:
        // verify(mockApi).sendData(anyString());
    }
}
```

```
}
```

```
}
```

```
MyService.java
```

```
package com.study;
```

```
public class MyService {
```

```
    private ExternalApi externalApi;
```

```
    public MyService(ExternalApi externalApi) {
```

```
        this.externalApi = externalApi;
```

```
    }
```

```
    public void process(String input) {
```

```
        externalApi.sendData(input);
```

```
    }
```

```
}
```

```
ExternalApi
```

```
package com.study;
```

```
public interface ExternalApi {
```

```
    String sendData(String message);
```

```
}
```

```
Pom.xml
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
    http://maven.apache.org/xsd/maven-4.0.0.xsd">  
    <modelVersion>4.0.0</modelVersion>
```



```
    <groupId>com.study</groupId>  
    <artifactId>Project</artifactId>  
    <version>1.0-SNAPSHOT</version>
```



```
    <properties>
```

```
        <maven.compiler.source>1.8</maven.compiler.source>
```

```
        <maven.compiler.target>1.8</maven.compiler.target>
```

```
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
```

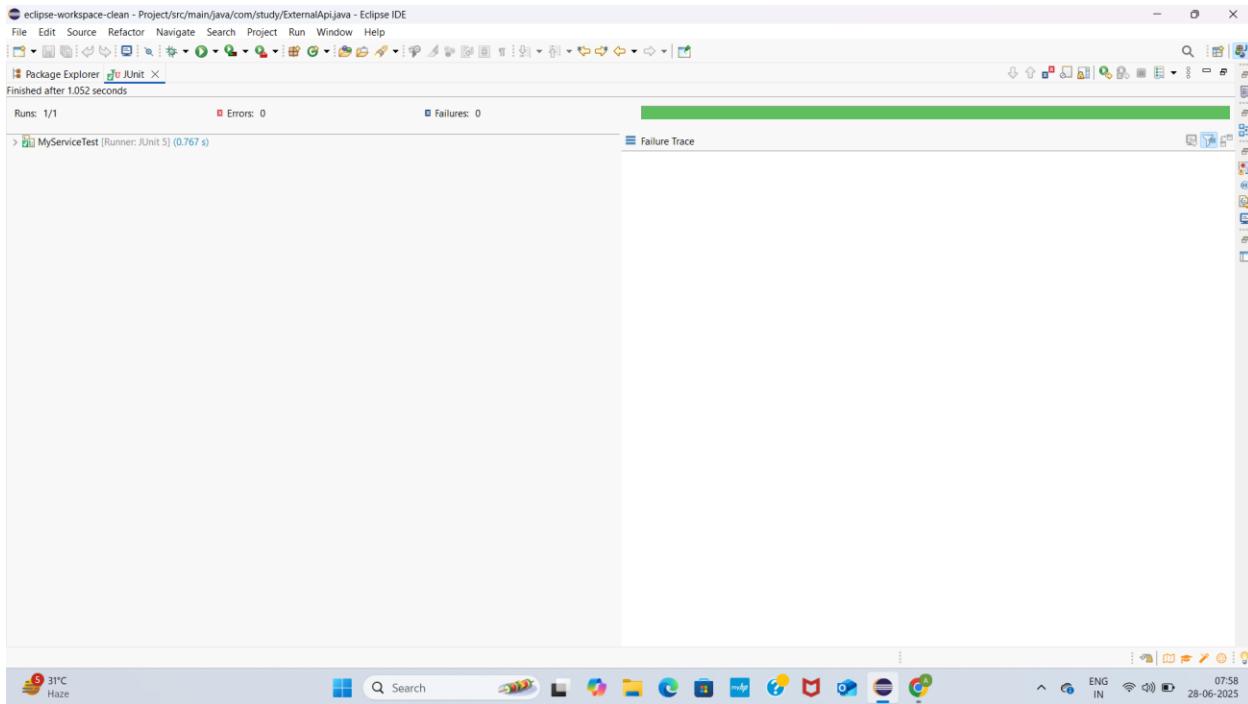
```
    </properties>
```

```
<dependencies>
    <!-- JUnit 5 -->
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter</artifactId>
        <version>5.10.0</version>
        <scope>test</scope>
    </dependency>

    <!-- Mockito -->
    <dependency>
        <groupId>org.mockito</groupId>
        <artifactId>mockito-core</artifactId>
        <version>5.12.0</version>
        <scope>test</scope>
    </dependency>
</dependencies>

<build>
    <plugins>
        <!-- Plugin to run JUnit 5 tests -->
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-surefire-plugin</artifactId>
            <version>3.1.2</version>
            <configuration>
                <useModulePath>false</useModulePath>
            </configuration>
        </plugin>
    </plugins>
</build>
</project>
```

OUTPUTSCREENSHOT



4.HANDLING VOID METHODS

```
MyServiceTest.java
package com.study;
import static org.mockito.Mockito.*;
import static org.mockito.ArgumentMatchers.*;
import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

    @Test
    public void testArgumentMatching() {
        // Step 1: Create mock
        ExternalApi mockApi = Mockito.mock(ExternalApi.class);

        // Step 2: Inject mock and call method with specific argument
        MyService service = new MyService(mockApi);
        service.performLogging();

        // Step 3: Verify it was called with specific argument using matcher
        verify(mockApi).log("Action performed");
        // You could also match any string like:
        // verify(mockApi).sendData(anyString());
    }
}
```

```
}
```

```
MyService.java
package com.study;

public class MyService {

    private ExternalApi externalApi;

    public MyService(ExternalApi externalApi) {
        this.externalApi = externalApi;
    }

    public void performLogging() {
        externalApi.log("Action performed");
    }
}
ExternalApi
package com.study;

public interface ExternalApi {
    void log(String message);
}
```

```
Pom.xml
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.study</groupId>
    <artifactId>Project</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

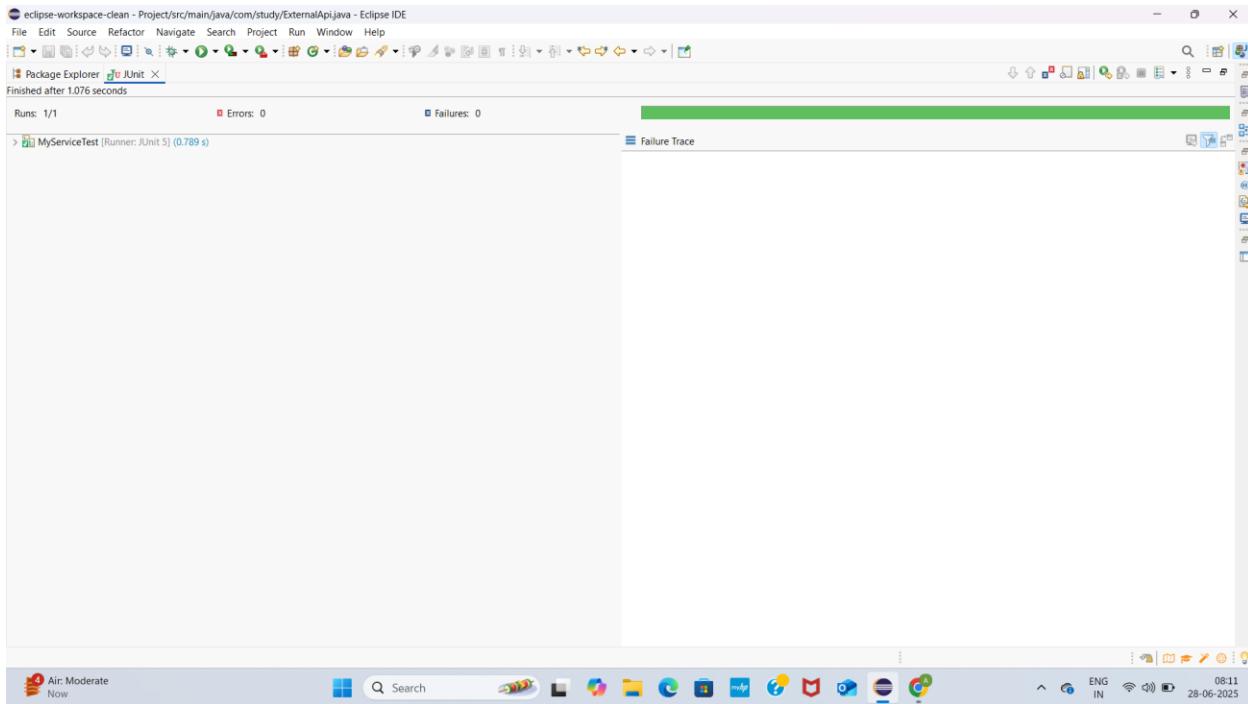
    <dependencies>
```

```
<!-- JUnit 5 -->
<dependency>
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter</artifactId>
<version>5.10.0</version>
<scope>test</scope>
</dependency>

<!-- Mockito -->
<dependency>
<groupId>org.mockito</groupId>
<artifactId>mockito-core</artifactId>
<version>5.12.0</version>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<!-- Plugin to run JUnit 5 tests -->
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>3.1.2</version>
<configuration>
<useModulePath>false</useModulePath>
</configuration>
</plugin>
</plugins>
</build>
</project>
```

OUTPUT SCREENSHOT



5. MOCKING AND STUBBING WITH MULTIPLE RETURNS

MyServiceTest.java

```
package com.study;

import static org.mockito.Mockito.*;
import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.api.Test;
import org.mockito.Mockito;

public class MyServiceTest {

    @Test
    public void testMultipleReturnValues() {
        // Step 1: Create mock object
        ExternalApi mockApi = Mockito.mock(ExternalApi.class);

        // Step 2: Stub method to return different values each time
        when(mockApi.getStatus())
            .thenReturn("Pending")
            .thenReturn("Processing")
            .thenReturn("Complete");

        // Step 3: Call method and assert results
        MyService service = new MyService(mockApi);
        String[] statuses = service.checkStatusMultipleTimes();
    }
}
```

```
assertArrayEquals(new String[] {"Pending", "Processing", "Complete"}, statuses);
}
}
```

MyService.java

```
package com.study;

public class MyService {

private ExternalApi externalApi;

public MyService(ExternalApi externalApi) {
this.externalApi = externalApi;
}

public String[] checkStatusMultipleTimes() {
return new String[] {
externalApi.getStatus(), // Call 1
externalApi.getStatus(), // Call 2
externalApi.getStatus() // Call 3
};
}
}
```

ExternalApi

```
package com.study;

public interface ExternalApi {
String getStatus();
}
```

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<groupId>com.study</groupId>
<artifactId>Project</artifactId>
<version>1.0-SNAPSHOT</version>
```

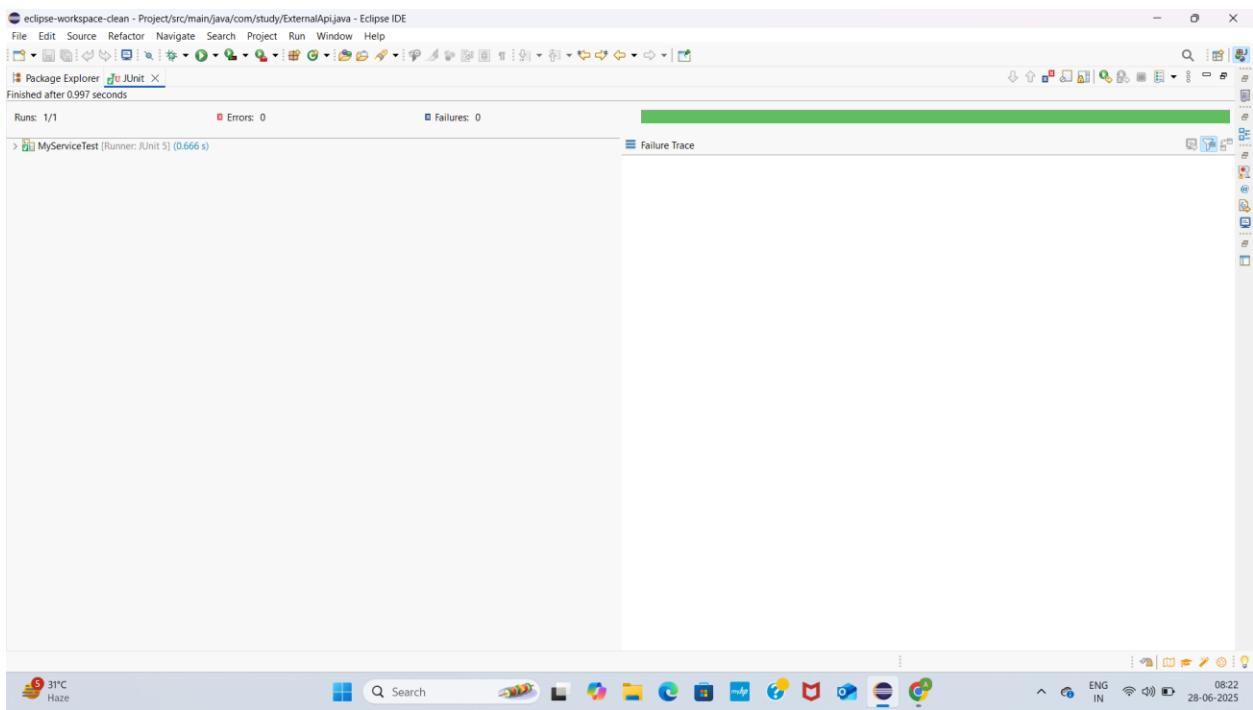
```
<properties>
<maven.compiler.source>1.8</maven.compiler.source>
<maven.compiler.target>1.8</maven.compiler.target>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>

<dependencies>
<!--  JUnit 5 -->
<dependency>
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter</artifactId>
<version>5.10.0</version>
<scope>test</scope>
</dependency>

<!--  Mockito -->
<dependency>
<groupId>org.mockito</groupId>
<artifactId>mockito-core</artifactId>
<version>5.12.0</version>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<!-- Plugin to run JUnit 5 tests -->
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>3.1.2</version>
<configuration>
<useModulePath>false</useModulePath>
</configuration>
</plugin>
</plugins>
</build>
</project>
```

OUTPUTSCREENSHOT



6. VERIFYING INTERACTION ORDER

MyServiceTest.java

```
package com.study;
```

```
import static org.mockito.Mockito.*;
```

```
import org.junit.jupiter.api.Test;  
import org.mockito.InOrder;
```

```
public class MyServiceTest {
```

@Test

```
public void testInteractionOrder() {
```

// Step 1: Create mock

```
ExternalApi mockApi = mock(ExternalApi.class);
```

// Step 2: Use the mock in service

```
MyService service = new MyService(mockApi);
```

```
service.execute(); // Calls: connect → send → disconnect
```

// Step 3: Verify call order

```
InOrder inOrder = inOrder(mockApi);
```

```
inOrder.verify(mockApi).connect();
```

```
inOrder.verify(mockApi).send("Hello");
```

```
inOrder.verify(mockApi).disconnect();
```

}

}

```
MyService.java
package com.study;

public class MyService {

    private ExternalApi externalApi;

    public MyService(ExternalApi externalApi) {
        this.externalApi = externalApi;
    }

    public void execute() {
        externalApi.connect();
        externalApi.send("Hello");
        externalApi.disconnect();
    }
}
```

```
ExternalApi
package com.study;

public interface ExternalApi {
    void connect();
    void send(String data);
    void disconnect();
}
```

```
Pom.xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.study</groupId>
    <artifactId>Project</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
```

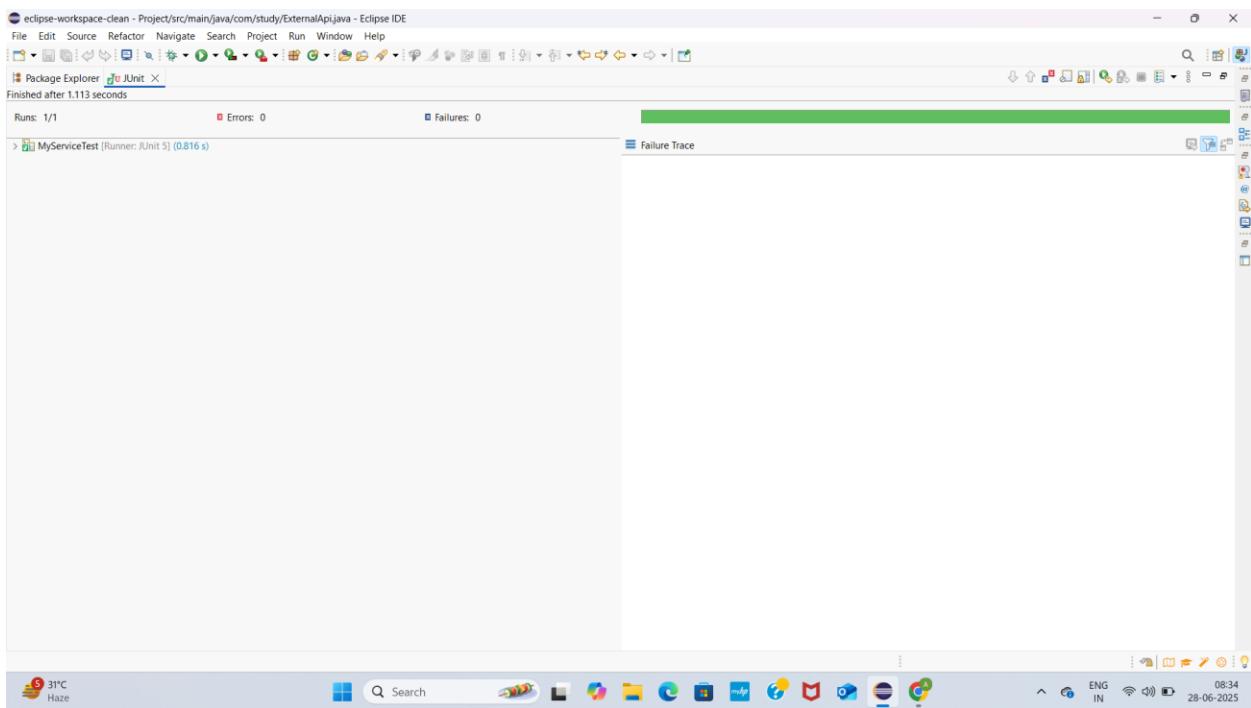
```
</properties>

<dependencies>
    <!-- JUnit 5 -->
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter</artifactId>
        <version>5.10.0</version>
        <scope>test</scope>
    </dependency>

    <!-- Mockito -->
    <dependency>
        <groupId>org.mockito</groupId>
        <artifactId>mockito-core</artifactId>
        <version>5.12.0</version>
        <scope>test</scope>
    </dependency>
</dependencies>

<build>
    <plugins>
        <!-- Plugin to run JUnit 5 tests -->
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-surefire-plugin</artifactId>
            <version>3.1.2</version>
            <configuration>
                <useModulePath>false</useModulePath>
            </configuration>
        </plugin>
    </plugins>
</build>
</project>
```

OUTPUT SCREENSHOT



7. HANDLING VOID METHODS

ExternalApi

```
package com.study;
```

```
public interface ExternalApi {  
    void deleteRecord(String id); // void method that may throw  
}
```

MyService.java

```
package com.study;
```

```
public class MyService {  
  
    private ExternalApi externalApi;  
  
    public MyService(ExternalApi externalApi) {  
        this.externalApi = externalApi;  
    }  
  
    public void remove(String id) {  
        externalApi.deleteRecord(id);  
    }  
}
```

MyServiceTest.java

```

package com.study;

import static org.mockito.Mockito.*;
import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.api.Test;
import org.mockito.Mockito;

public class MyServiceTest {

    @Test
    public void testVoidMethodThrowsException() {
        // Step 1: Create mock
        ExternalApi mockApi = Mockito.mock(ExternalApi.class);

        // Step 2: Stub the void method to throw exception
        doThrow(new RuntimeException("Delete failed"))
            .when(mockApi).deleteRecord("123");

        // Step 3: Create service and call the method (should throw)
        MyService service = new MyService(mockApi);

        // Verify the exception is thrown
        RuntimeException ex = assertThrows(RuntimeException.class, () -> {
            service.remove("123");
        });

        assertEquals("Delete failed", ex.getMessage());

        // Step 4: Verify the interaction occurred
        verify(mockApi).deleteRecord("123");
    }
}

```

Pom.xml

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.study</groupId>
    <artifactId>Project</artifactId>
    <version>1.0-SNAPSHOT</version>

```

```

<properties>
<maven.compiler.source>1.8</maven.compiler.source>
<maven.compiler.target>1.8</maven.compiler.target>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>

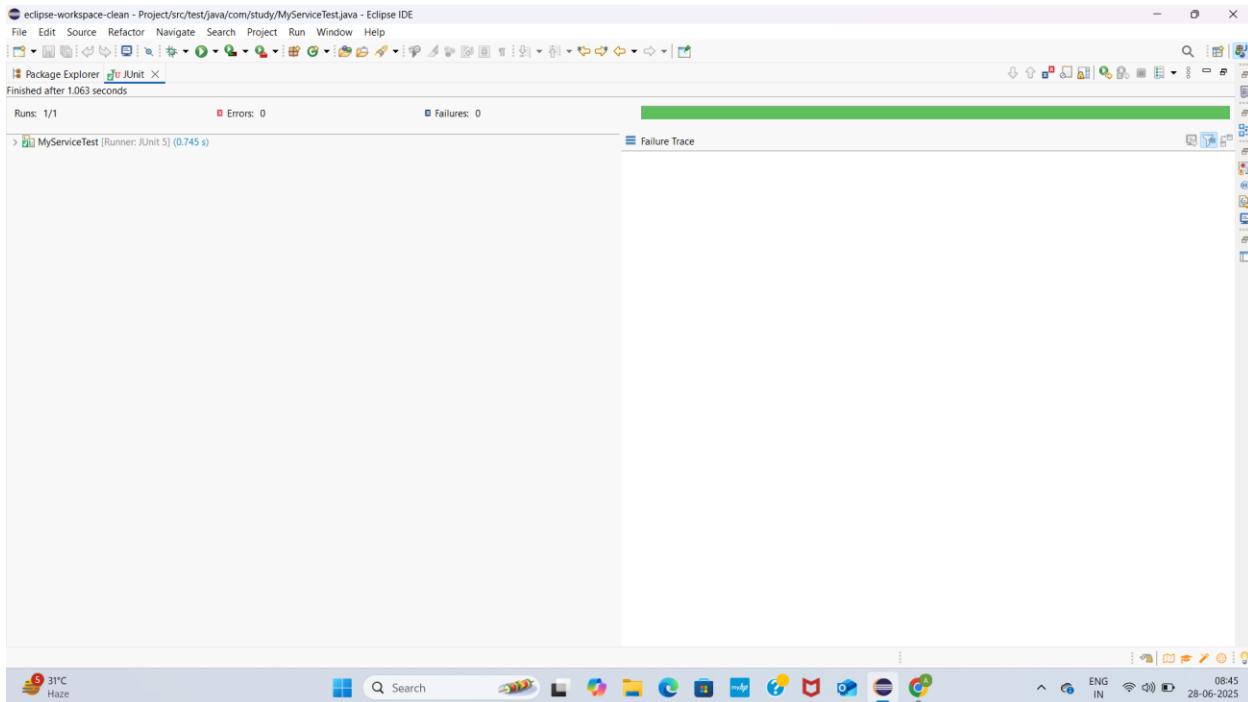
<dependencies>
<!-- JUnit 5 -->
<dependency>
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter</artifactId>
<version>5.10.0</version>
<scope>test</scope>
</dependency>

<!-- Mockito -->
<dependency>
<groupId>org.mockito</groupId>
<artifactId>mockito-core</artifactId>
<version>5.12.0</version>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<!-- Plugin to run JUnit 5 tests -->
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>3.1.2</version>
<configuration>
<useModulePath>false</useModulePath>
</configuration>
</plugin>
</plugins>
</build>
</project>

```

OUTPUTSCREENSHOT



ADVANCED MOCKITO HANDS ON

1.

Repository.java

```
package com.study;
```

```
public interface Repository {  
    String getData();
```

```
}
```

Service.java

```
package com.study;
```

```
public class Service {  
    private Repository repository;
```

```
    public Service(Repository repository) {  
        this.repository = repository;  
    }
```

```
    public String processData() {  
        String data = repository.getData(); // gets "Mock Data"  
        return "Processed " + data; // appends logic  
    }  
}
```

```
Servicetest.java
package com.study;
import static org.mockito.Mockito.*;
import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.api.Test;

public class ServiceTest {

    @Test
    public void testServiceWithMockRepository() {
        // Step 1: Create a mock repository
        Repository mockRepository = mock(Repository.class);

        // Step 2: Stub method to return mock data
        when(mockRepository.getData()).thenReturn("Mock Data");

        // Step 3: Inject into service and test logic
        Service service = new Service(mockRepository);
        String result = service.processData();

        // Step 4: Verify result
        assertEquals("Processed Mock Data", result);
    }
}
```

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.study</groupId>
    <artifactId>Project</artifactId>
    <version>1.0-SNAPSHOT</version>

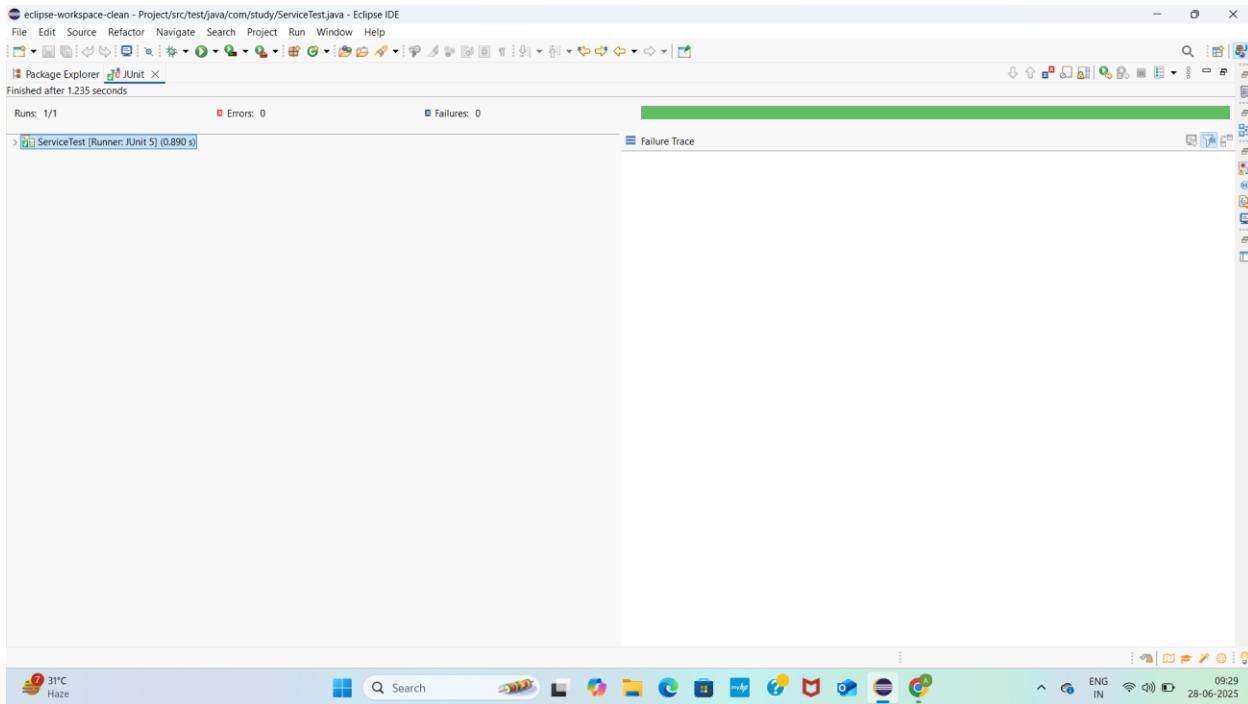
    <properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>
```

```
<dependencies>
  <!-- JUnit 5 -->
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter</artifactId>
    <version>5.10.0</version>
    <scope>test</scope>
  </dependency>

  <!-- Mockito -->
  <dependency>
    <groupId>org.mockito</groupId>
    <artifactId>mockito-core</artifactId>
    <version>5.12.0</version>
    <scope>test</scope>
  </dependency>
</dependencies>

<build>
  <plugins>
    <!-- Plugin to run JUnit 5 tests -->
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>3.1.2</version>
      <configuration>
        <useModulePath>false</useModulePath>
      </configuration>
    </plugin>
  </plugins>
</build>
</project>
```

OUTPUT SCREENSHOT



```
2. package com.study;
```

```
public interface RestClient {
```

```
String getResponse();
```

```
}
```

```
ApiService.java
```

```
package com.study;
```

```
public class ApiService {
```

```
private RestClient restClient;
```

```
public ApiService(RestClient restClient) {
```

```
this.restClient = restClient;
```

```
}
```

```
public String fetchData() {
```

```
String response = restClient.getResponse(); // calls mocked REST API
```

```
return "Fetched " + response; // processes result
```

```
}
```

```
}
```

```
ApiServiceTest.java
```

```
package com.study;
```

```

import static org.mockito.Mockito.*;
import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.api.Test;

public class ApiServiceTest {

    @Test
    public void testServiceWithMockRestClient() {
        // Step 1: Create mock REST client
        RestClient mockRestClient = mock(RestClient.class);

        // Step 2: Stub the response
        when(mockRestClient.getResponse()).thenReturn("Mock Response");

        // Step 3: Inject into service and test
        ApiService apiService = new ApiService(mockRestClient);
        String result = apiService.fetchData();

        // Step 4: Verify logic
        assertEquals("Fetched Mock Response", result);
    }
}

```

Pom.xml

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.study</groupId>
    <artifactId>Project</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>
        <!-- JUnit 5 -->
        <dependency>

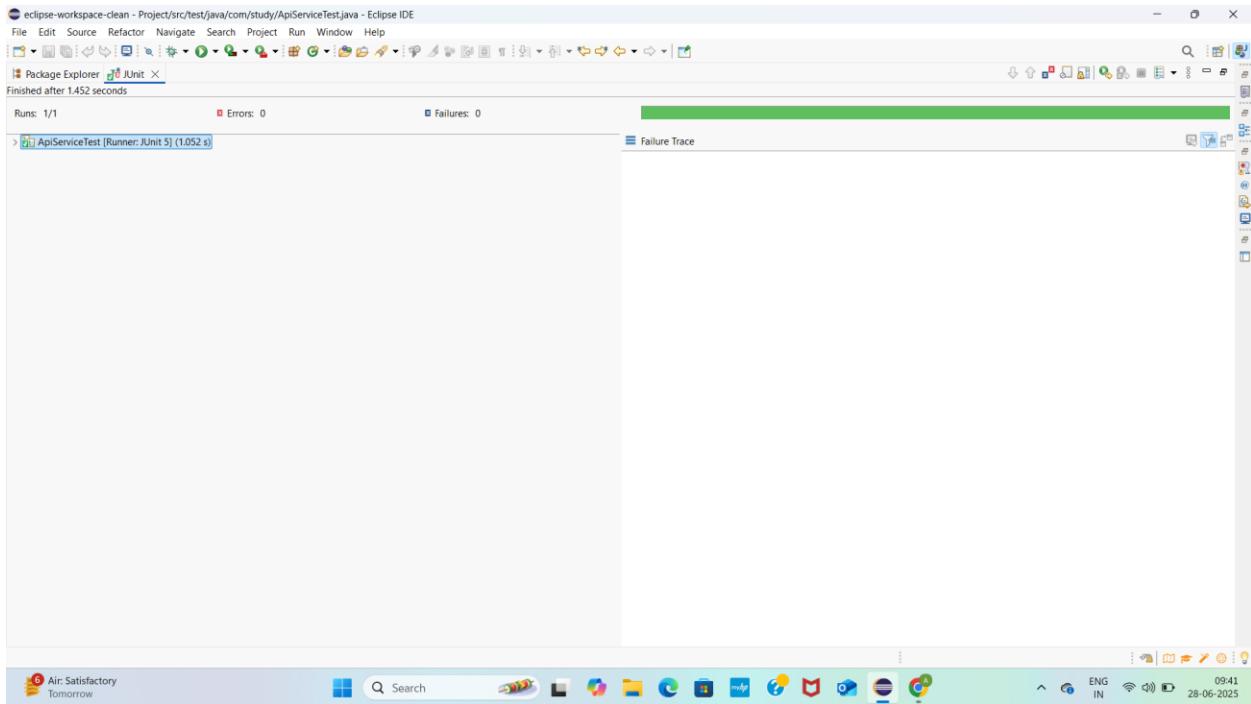
```

```
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter</artifactId>
<version>5.10.0</version>
<scope>test</scope>
</dependency>

<!--  Mockito -->
<dependency>
<groupId>org.mockito</groupId>
<artifactId>mockito-core</artifactId>
<version>5.12.0</version>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<!-- Plugin to run JUnit 5 tests -->
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>3.1.2</version>
<configuration>
<useModulePath>false</useModulePath>
</configuration>
</plugin>
</plugins>
</build>
</project>
```

OUTPUTSCREENSHOT



3. FileReader.java

```
package com.study;

public interface FileReader {
    String read();
}
```

FileWriter.java

```
package com.study;

public interface FileWriter {
    void write(String data);
}
```

FileService.java

```
package com.study;

public class FileService {
    private FileReader fileReader;
    private FileWriter fileWriter;

    public FileService(FileReader fileReader, FileWriter fileWriter) {
```

```
this.FileReader = fileReader;
this.FileWriter = FileWriter;
}

public String processFile() {
String content = FileReader.read(); // Simulate reading file
String processed = "Processed " + content;
FileWriter.write(processed); // Simulate writing file
return processed;
}
}
```

FileServiceTest.java

```
package com.study;
import static org.mockito.Mockito.*;
import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.api.Test;

public class FileServiceTest {

@Test
public void testServiceWithMockFileIO() {
// Step 1: Create mock FileReader and FileWriter
FileReader mockFileReader = mock(FileReader.class);
FileWriter mockFileWriter = mock(FileWriter.class);

// Step 2: Stub the FileReader to return mock content
when(mockFileReader.read()).thenReturn("Mock File Content");

// Step 3: Create the service with mocks
FileService fileService = new FileService(mockFileReader, mockFileWriter);

// Step 4: Call the service method
String result = fileService.processFile();

// Step 5: Verify result and that FileWriter was called correctly
assertEquals("Processed Mock File Content", result);
verify(mockFileWriter).write("Processed Mock File Content"); // Interaction verification
}

}

Pom.xml
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.study</groupId>
  <artifactId>Project</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>

  <dependencies>
    <!--  JUnit 5 -->
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter</artifactId>
      <version>5.10.0</version>
      <scope>test</scope>
    </dependency>

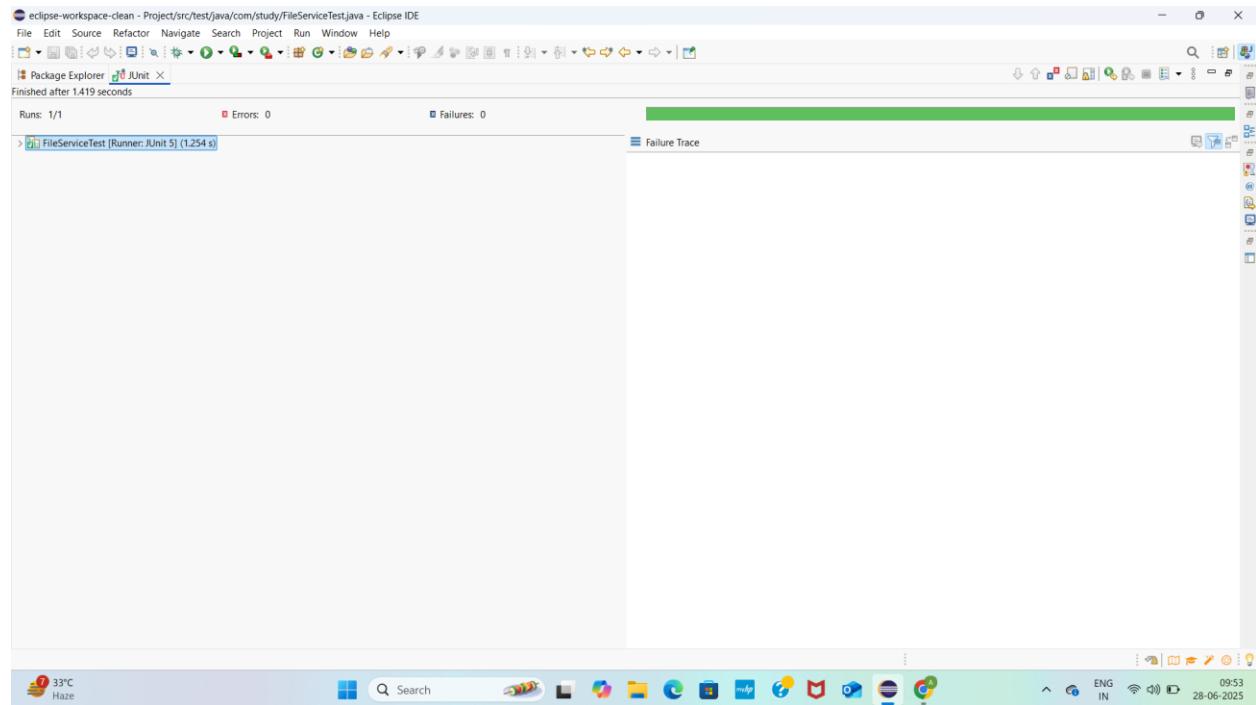
    <!--  Mockito -->
    <dependency>
      <groupId>org.mockito</groupId>
      <artifactId>mockito-core</artifactId>
      <version>5.12.0</version>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <!-- Plugin to run JUnit 5 tests -->
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-surefire-plugin</artifactId>
        <version>3.1.2</version>
        <configuration>
          <useModulePath>false</useModulePath>
        </configuration>
      </plugin>
    </plugins>
  </build>

```

```
</build>
</project>
```

OUTPUT SCREENSHOT



4.

```
NetworkClient.java
package com.study;
```

```
public interface NetworkClient {
String connect();

}
```

```
NetworkService.java
```

```
package com.study;

public class NetworkService {
private NetworkClient networkClient;

public NetworkService(NetworkClient networkClient) {
this.networkClient = networkClient;
}

public String connectToServer() {
String connection = networkClient.connect(); // simulate a real connection
}
```

```
        return "Connected to " + connection; // service logic
    }
}
```

NetworkServiceTest.java

```
package com.study;
import static org.mockito.Mockito.*;
import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.api.Test;

public class NetworkServiceTest {

    @Test
    public void testServiceWithMockNetworkClient() {
        // Step 1: Create mock network client
        NetworkClient mockNetworkClient = mock(NetworkClient.class);

        // Step 2: Stub the connect method
        when(mockNetworkClient.connect()).thenReturn("Mock Connection");

        // Step 3: Inject into service and test
        NetworkService networkService = new NetworkService(mockNetworkClient);
        String result = networkService.connectToServer();

        // Step 4: Verify output
        assertEquals("Connected to Mock Connection", result);

        // Optional: Verify interaction
        verify(mockNetworkClient).connect();
    }
}
```

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.study</groupId>
    <artifactId>Project</artifactId>
    <version>1.0-SNAPSHOT</version>
```

```

<properties>
<maven.compiler.source>1.8</maven.compiler.source>
<maven.compiler.target>1.8</maven.compiler.target>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>

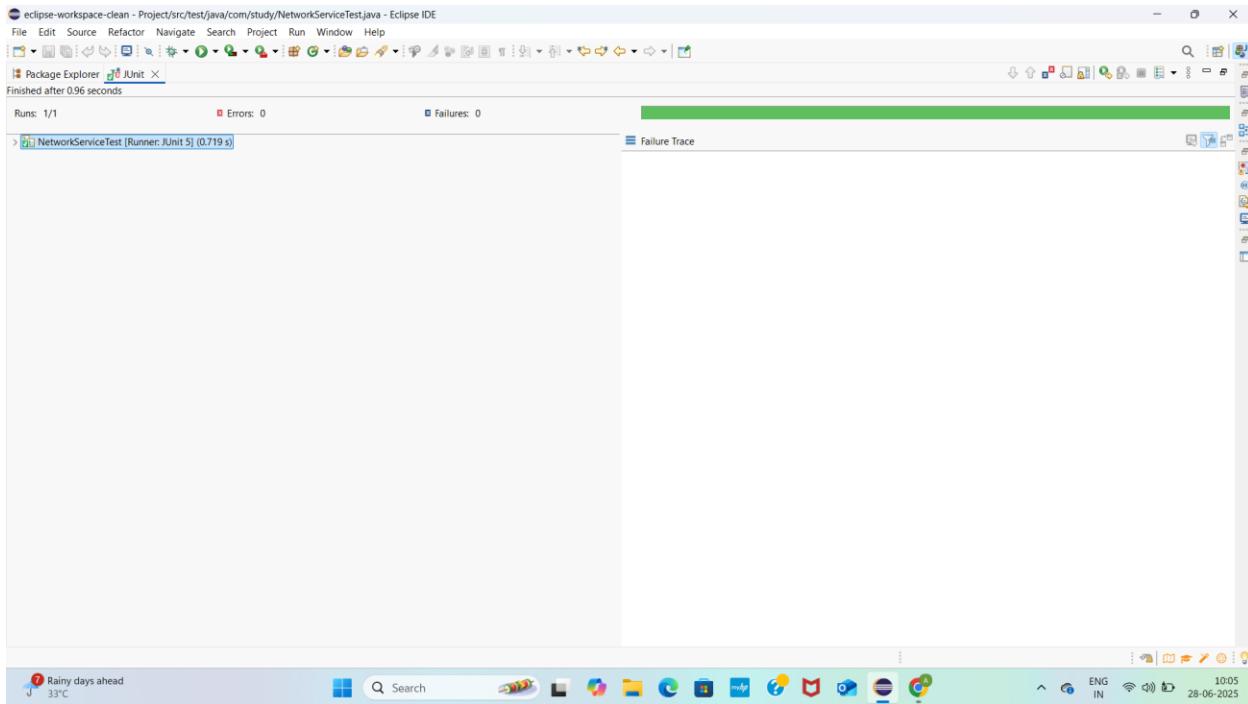
<dependencies>
<!--  JUnit 5 -->
<dependency>
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter</artifactId>
<version>5.10.0</version>
<scope>test</scope>
</dependency>

<!--  Mockito -->
<dependency>
<groupId>org.mockito</groupId>
<artifactId>mockito-core</artifactId>
<version>5.12.0</version>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<!-- Plugin to run JUnit 5 tests -->
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>3.1.2</version>
<configuration>
<useModulePath>false</useModulePath>
</configuration>
</plugin>
</plugins>
</build>
</project>

```

OUTPUTSCREENSHOT



5.

Repository.java

```
package com.study;
```

```
public interface Repository {  
    String getData();
```

```
}
```

Service.java

```
package com.study;
```

```
public class Service {  
    private Repository repository;
```

```
    public Service(Repository repository) {  
        this.repository = repository;  
    }
```

```
    public String processData() {  
        String data = repository.getData(); // gets "Mock Data"  
        return "Processed " + data; // appends logic  
    }  
}
```

MultiReturnServiceTest.java

```

package com.study;
import static org.mockito.Mockito.*;
import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.api.Test;

public class MultiReturnServiceTest {

    @Test
    public void testServiceWithMultipleReturnValues() {
        // Step 1: Create mock
        Repository mockRepository = mock(Repository.class);

        // Step 2: Stub with multiple return values
        when(mockRepository.getData())
            .thenReturn("First Mock Data")
            .thenReturn("Second Mock Data");

        // Step 3: Inject into service and call twice
        Service service = new Service(mockRepository);
        String firstResult = service.processData();
        String secondResult = service.processData();

        // Step 4: Verify both results
        assertEquals("Processed First Mock Data", firstResult);
        assertEquals("Processed Second Mock Data", secondResult);

        // Optional: Verify interaction count
        verify(mockRepository, times(2)).getData();
    }
}

```

Pom.xml

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                             http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.study</groupId>
    <artifactId>Project</artifactId>
    <version>1.0-SNAPSHOT</version>

```

```

<properties>
<maven.compiler.source>1.8</maven.compiler.source>
<maven.compiler.target>1.8</maven.compiler.target>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>

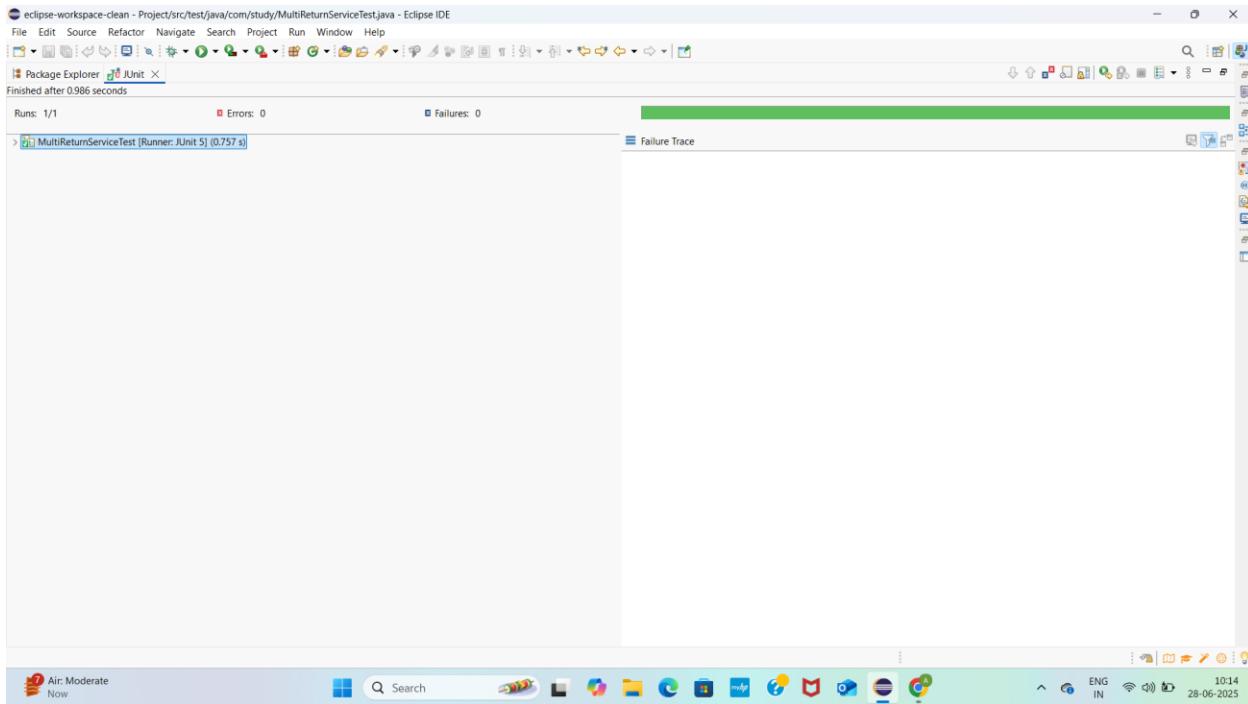
<dependencies>
<!-- JUnit 5 -->
<dependency>
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter</artifactId>
<version>5.10.0</version>
<scope>test</scope>
</dependency>

<!-- Mockito -->
<dependency>
<groupId>org.mockito</groupId>
<artifactId>mockito-core</artifactId>
<version>5.12.0</version>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<!-- Plugin to run JUnit 5 tests -->
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>3.1.2</version>
<configuration>
<useModulePath>false</useModulePath>
</configuration>
</plugin>
</plugins>
</build>
</project>

```

OUTPUT SCREENSHOT



SPRING TESTING EXERCISE

1.

CalculatorService.java

```
package com.study;
import org.springframework.stereotype.Service;

@Service
public class CalculatorService {
    public int add(int a, int b) {
        return a + b;
    }
}
```

CalculatorServiceTest.java

```
package com.study;
```

```
import static org.junit.jupiter.api.Assertions.*;
```

```
import org.junit.jupiter.api.Test;
```

```
public class CalculatorServiceTest {
```

```
@Test
```

```
public void testAdd() {
```

```
    CalculatorService calculatorService = new CalculatorService();
```

```
    int result = calculatorService.add(5, 3);
```

```
    assertEquals(8, result, "5 + 3 should equal 8");
```

```
}

}

Pom.xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.study</groupId>
  <artifactId>spring-junit-example</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>
    <java.version>17</java.version>
    <spring.version>6.1.6</spring.version>
    <junit.version>5.10.0</junit.version>
    <mockito.version>5.12.0</mockito.version>
    <maven.compiler.source>${java.version}</maven.compiler.source>
    <maven.compiler.target>${java.version}</maven.compiler.target>
  </properties>

  <dependencies>
    <!-- Spring Core (for @Service, etc.) -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>${spring.version}</version>
    </dependency>

    <!-- JUnit 5 -->
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter</artifactId>
      <version>${junit.version}</version>
      <scope>test</scope>
    </dependency>

    <!-- Mockito -->
    <dependency>
      <groupId>org.mockito</groupId>
      <artifactId>mockito-core</artifactId>
      <version>${mockito.version}</version>
      <scope>test</scope>
    </dependency>
  
```

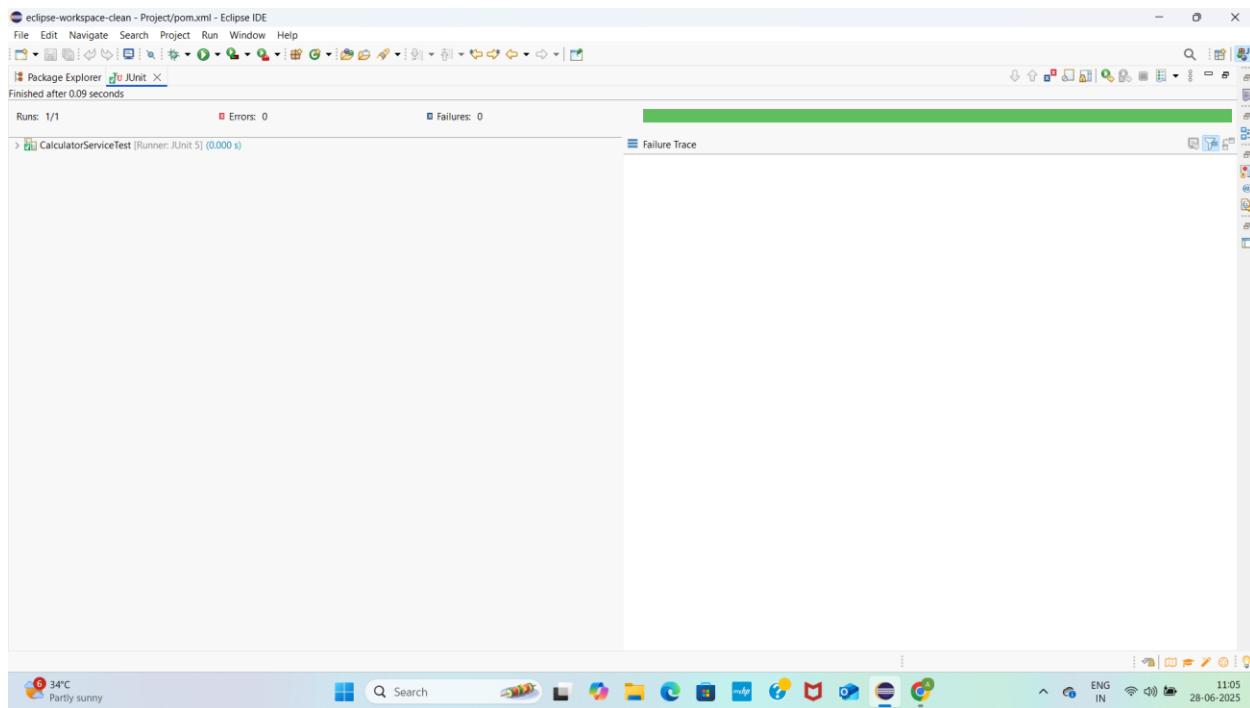
```

</dependencies>

<build>
<plugins>
<!-- Use Maven Surefire Plugin for JUnit 5 -->
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>3.2.5</version>
<configuration>
<useModulePath>false</useModulePath>
</configuration>
</plugin>
</plugins>
</build>
</project>

```

OUTPUTSCREENSHOT



MOCKING DEPENDENCIES IN SPRING TESTS USING MOCKITO

1.

UserControllerTest.java

```

package com.study;
import org.junit.jupiter.api.Test;

```

```
import org.springframework.boot.test.autoconfigure.web.servlet.WebMvcTest;
import org.springframework.boot.test.mock.mockito.MockBean;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.test.web.servlet.MockMvc;

import static org.mockito.Mockito.when;
import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.get;
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.*;

@WebMvcTest(UserController.class)
public class UserControllerTest {

    @Autowired
    private MockMvc mockMvc;

    @MockBean
    private UserService userService;

    @Test
    public void testGetUserById() throws Exception {
        when(userService.getUserById(1L)).thenReturn("Mocked User");

        mockMvc.perform(get("/users/1"))
            .andExpect(status().isOk())
            .andExpect(content().string("Mocked User"));
    }
}
```

```
UserController.java
package com.study;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/users")
public class UserController {

    @Autowired
    private UserService userService;
```

```
@GetMapping("/{id}")
public ResponseEntity<String> getUser(@PathVariable Long id) {
    return ResponseEntity.ok(userService.getUserById(id));
}
}
```

UserService.java
package com.study;

```
public interface UserService {
    String getUserById(Long id);
}
```

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```
<modelVersion>4.0.0</modelVersion>
```

```
<parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.2.5</version>
    <relativePath/>
</parent>
```

```
<groupId>com.study</groupId>
<artifactId>spring-basic</artifactId>
<version>1.0.0</version>
```

```
<properties>
    <java.version>17</java.version>
</properties>
```

```
<dependencies>
    <!-- ☑ Spring Web -->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
```

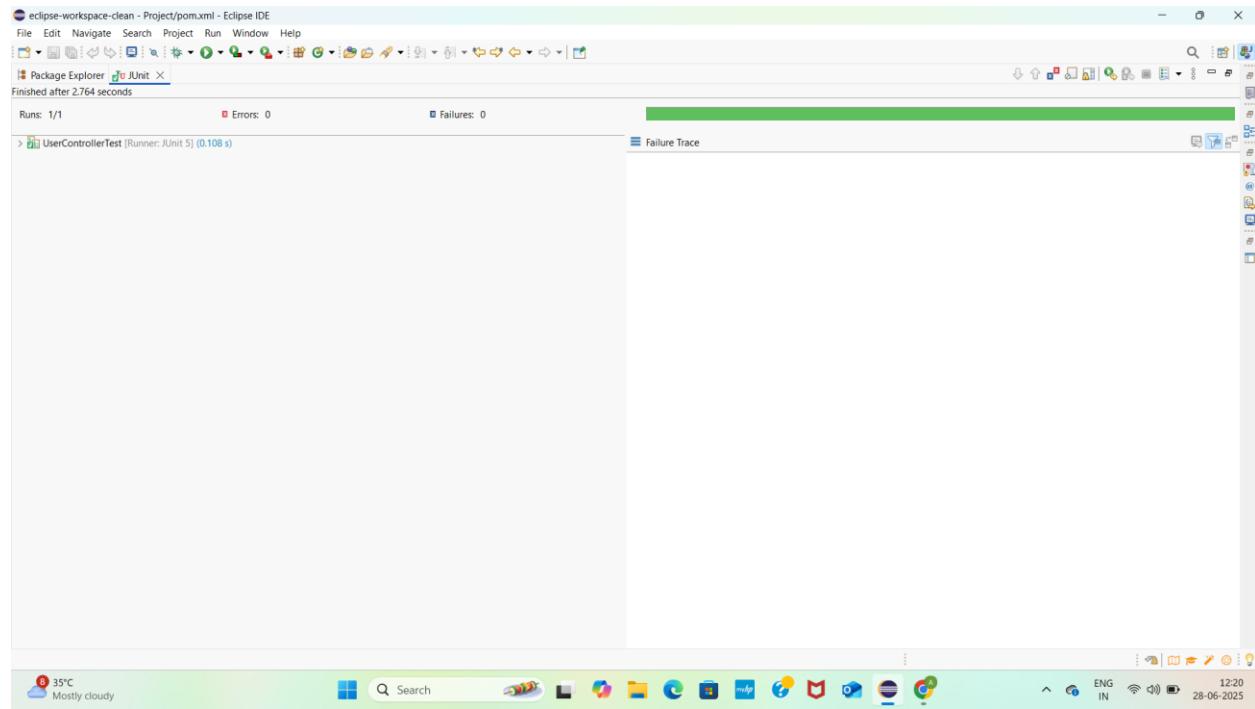
```

<!-- JUnit 5 + Mockito -->
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<!-- Spring Boot Plugin -->
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
</project>

```

OUTPUTSCREENSHOT



2.

User.java

```
package com.study;
```

```
public class User {  
    private Long id;  
    private String name;  
  
    public User(Long id, String name) {  
        this.id = id;  
        this.name = name;  
    }  
  
    public Long getId() {  
        return id;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```

```
UserRepository.java  
package com.study;  
  
import java.util.HashMap;  
import java.util.Map;  
import java.util.Optional;  
  
public class UserRepository {  
    private static final Map<Long, User> store = new HashMap<>();  
  
    static {  
        store.put(1L, new User(1L, "Alice"));  
        store.put(2L, new User(2L, "Bob"));  
    }  
  
    public Optional<User> findById(Long id) {  
        return Optional.ofNullable(store.get(id));  
    }  
}
```

```
UserService.java  
package com.study;  
  
public class UserService {  
  
    private final UserRepository userRepository;
```

```
public UserService(UserRepository userRepository) {
    this.userRepository = userRepository;
}

public User getUserById(Long id) {
    return userRepository.findById(id).orElse(null);
}
}
```

UserServiceTest.java

```
package com.study;

import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class UserServiceTest {

    @Test
    public void test GetUserById_Found() {
        UserRepository repo = new UserRepository();
        UserService service = new UserService(repo);

        User user = service.getUserById(1L);
        assertNotNull(user);
        assertEquals("Alice", user.getName());
    }

    @Test
    public void test GetUserById_NotFound() {
        UserRepository repo = new UserRepository();
        UserService service = new UserService(repo);

        User user = service.getUserById(99L);
        assertNull(user);
    }
}

Pom.xml

<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```
<modelVersion>4.0.0</modelVersion>

<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>3.2.5</version>
<relativePath/>
</parent>

<groupId>com.study</groupId>
<artifactId>spring-basic</artifactId>
<version>1.0.0</version>

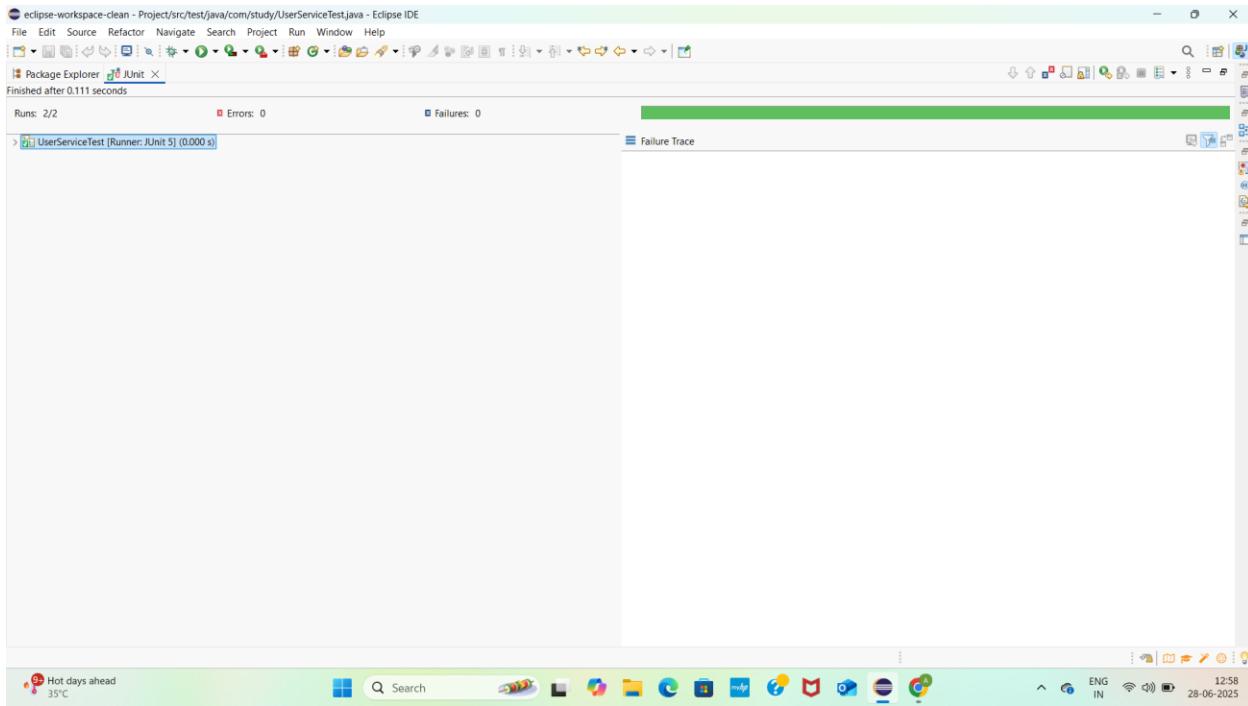
<properties>
<java.version>17</java.version>
</properties>

<dependencies>
<!--  Spring Web -->
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>

<!--  JUnit 5 + Mockito -->
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<!--  Spring Boot Plugin -->
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
</project>
```

OUTPUT SCREENSHOT



3. UserController.java

```
package com.study;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

@RestController
@RequestMapping("/users")
public class UserController {

    @Autowired
    private UserService userService;

    @GetMapping("/{id}")
    public ResponseEntity<User> getUser(@PathVariable Long id) {
        User user = userService.getUserById(id);
        if (user != null) {
            return ResponseEntity.ok(user);
        } else {
            return ResponseEntity.notFound().build();
        }
    }
}
```

```
User.java
package com.study;

public class User {
private Long id;
private String name;

public User(Long id, String name) {
this.id = id;
this.name = name;
}

public Long getId() {
return id;
}

public String getName() {
return name;
}
}
```

```
UserRepository.java
package com.study;

import java.util.HashMap;
import java.util.Map;
import java.util.Optional;

public class UserRepository {
private static final Map<Long, User> store = new HashMap<>();

static {
store.put(1L, new User(1L, "Alice"));
store.put(2L, new User(2L, "Bob"));
}

public Optional<User> findById(Long id) {
return Optional.ofNullable(store.get(id));
}
}
```

```
UserService.java
package com.study;
```

```
import org.springframework.stereotype.Service;

@Service
public class UserService {
    private final UserRepository userRepository = new UserRepository(); // Manual
    injection for simplicity

    public User getUserById(Long id) {
        return userRepository.findById(id).orElse(null);
    }
}
```

```
SpringApp.java
package com.study;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
@SpringBootApplication
public class SpringApp {
    public static void main(String[] args) {
        SpringApplication.run(SpringApp.class, args);
    }
}
```

```
UserControllerTest.java
package com.study;
import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.autoconfigure.web.servlet.WebMvcTest;
import org.springframework.boot.test.mock.mockito.MockBean;
import org.springframework.test.web.servlet.MockMvc;

import static org.mockito.Mockito.when;
import static
org.springframework.test.web.servlet.request.MockMvcBuilders.get;
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.*;

@WebMvcTest(UserController.class)
public class UserControllerTest {

    @Autowired
    private MockMvc mockMvc;

    @MockBean
```

```

private UserService userService;

@Test
public void test GetUser_Found() throws Exception {
when(userService.getUserById(1L)).thenReturn(new User(1L, "Alice"));

mockMvc.perform(get("/users/1"))
.andExpect(status().isOk())
.andExpect(jsonPath("$.name").value("Alice"));
}

@Test
public void test GetUser_NotFound() throws Exception {
when(userService.getUserById(99L)).thenReturn(null);

mockMvc.perform(get("/users/99"))
.andExpect(status().isNotFound());
}
}

```

Pom.xml

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>3.2.5</version>
<relativePath/>
</parent>

<groupId>com.study</groupId>
<artifactId>spring-basic</artifactId>
<version>1.0.0</version>

<properties>
<java.version>17</java.version>
</properties>

```

```
<dependencies>
    <!-- ☑ Spring Web -->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <!-- ☑ JUnit 5 + Mockito -->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>

<build>
    <plugins>
        <!-- ☑ Spring Boot Plugin -->
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
</project>
```

OUTPUTSCREENSHOT

eclipse-workspace-clean - Project:pom.xml - Eclipse IDE

File Edit Navigate Search Project Run Window Help

Problems Javadoc Declaration Console X

terminated> SpringApp [Java Application] C:\Users\abarna\p2\pool\plugins\org.eclipse.jst.openjdk.hotspot.jre.full.win32.x86_64_23.0.2.v20250131-0604\jre\bin\javaw.exe (28 Jun 2025, 20:51:00 pm - 20:51:20 pm elapsed: 0:00:02:482) [pid:7772]

Spring Boot :: (v3.2.5)

```
2025-06-28T14:05:10.989+05:30 INFO 7772 --- [main] com.study.SpringApp : Starting SpringApp using Java 23.0.2 with PID 7772 (C:\Users\abarna\eclipse-workspace-clean\Project)
2025-06-28T14:05:10.992+05:30 INFO 7772 --- [main] com.study.SpringApp : No active profile set, falling back to 1 default profile: "default"
2025-06-28T14:05:12.154+05:30 INFO 7772 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2025-06-28T14:05:12.172+05:30 INFO 7772 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-06-28T14:05:12.172+05:30 INFO 7772 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.20]
2025-06-28T14:05:12.233+05:30 INFO 7772 --- [main] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring embedded WebApplicationContext
2025-06-28T14:05:12.234+05:30 INFO 7772 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1186 ms
2025-06-28T14:05:12.571+05:30 WARN 7772 --- [main] ConfigServletWebServerApplicationContext : Exception encountered during context initialization - cancelling refresh attempt: org.springframework.web.context.ContextLoaderListener@...
2025-06-28T14:05:12.592+05:30 INFO 7772 --- [main] .s.b.a.l.ConditionEvaluationReporter :
```

Error starting ApplicationContext. To display the condition evaluation report re-run your application with 'debug' enabled.

```
2025-06-28T14:05:12.610+05:30 ERROR 7772 --- [main] o.s.b.d.LoggingFailureAnalysisReporter :
```

APPLICATION FAILED TO START

Description:

Web server failed to start. Port 8080 was already in use.

Action:

Identify and stop the process that's listening on port 8080 or configure this application to listen on another port.

4.

```
UserService.java
package com.study;

import org.springframework.stereotype.Service;

@Service
public class UserService {

    private final UserRepository userRepository;

    //  Constructor for Spring to inject UserRepository
    public UserService(UserRepository userRepository) {
        this.userRepository = userRepository;
    }
}
```

```
UserRepository.java  
package com.study;  
  
import org.springframework.stereotype.Repository;
```

```
import java.util.HashMap;
import java.util.Map;
import java.util.Optional;

@Repository
public class UserRepository {
private static final Map<Long, User> users = new HashMap<>();

static {
users.put(1L, new User(1L, "Alice"));
users.put(2L, new User(2L, "Bob"));
}

public Optional<User> findById(Long id) {
return Optional.ofNullable(users.get(id));
}
}
```

```
UserController.java
package com.study;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

@RestController
@RequestMapping("/users")
public class UserController {

private final UserService userService;

// ✅ Constructor-based injection
public UserController(UserService userService) {
this.userService = userService;
}

@GetMapping("/{id}")
public ResponseEntity<User> getUser(@PathVariable Long id) {
User user = userService.getUserById(id);
if (user != null) {
return ResponseEntity.ok(user);
} else {
return ResponseEntity.notFound().build();
}
}
```

```
}
```

```
}
```

```
}
```

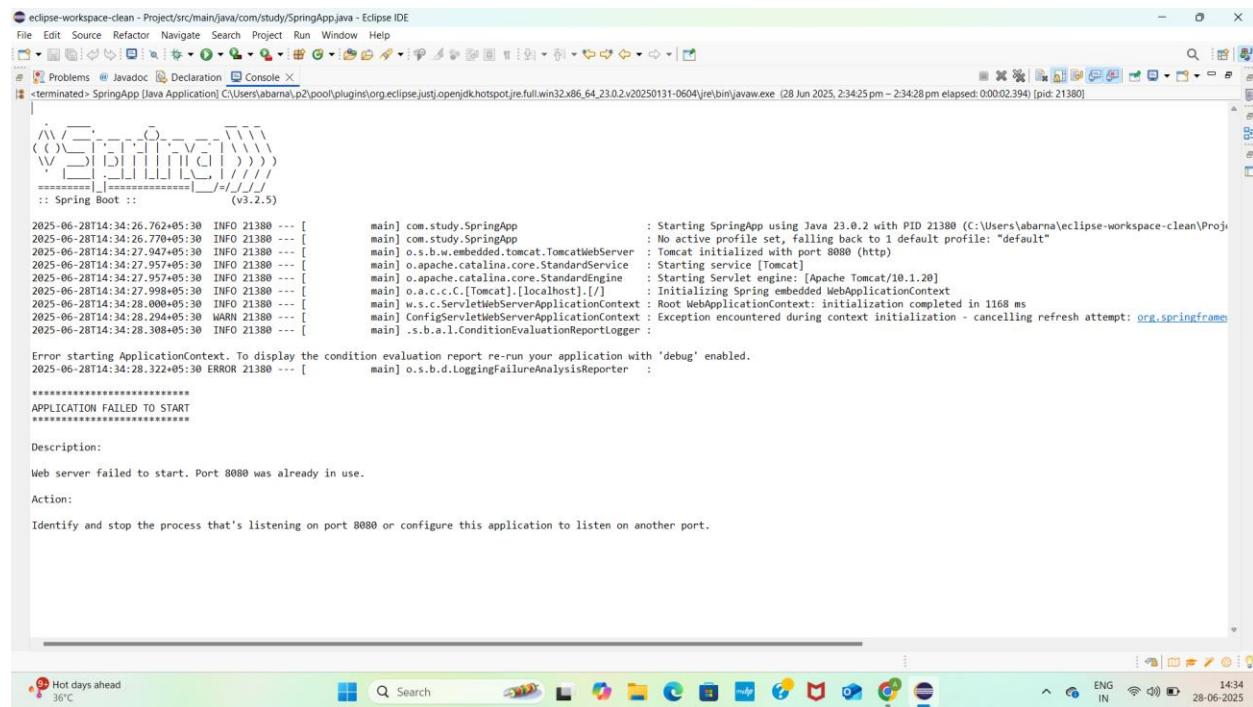
SpringApp.java

```
package com.study;
```

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
@SpringBootApplication
public class SpringApp {
    public static void main(String[] args) {
        SpringApplication.run(SpringApp.class, args);
    }
}
```

OUTPUTSCREENSHOT



The screenshot shows the Eclipse IDE interface with the SpringApp.java file open in the editor. The Java Console tab is selected, displaying the following log output:

```
2025-06-28T14:34:26.762+05:30 INFO 21380 --- [main] com.study.SpringApp : Starting SpringApp using Java 23.0.2 with PID 21380 (C:\Users\abarna\eclipse-workspace-clean\Proj)
2025-06-28T14:34:26.770+05:30 INFO 21380 --- [main] com.study.SpringApp : No active profile set, falling back to 1 default profile: "default"
2025-06-28T14:34:27.947+05:30 INFO 21380 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2025-06-28T14:34:27.957+05:30 INFO 21380 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-06-28T14:34:27.957+05:30 INFO 21380 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.20]
2025-06-28T14:34:27.998+05:30 INFO 21380 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2025-06-28T14:34:28.000+05:30 INFO 21380 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1168 ms
2025-06-28T14:34:28.294+05:30 WARN 21380 --- [main] ConfigServletWebServerApplicationContext : Exception encountered during context initialization - cancelling refresh attempt: org.springframework.web.context.ContextualEvaluationReportLogger
2025-06-28T14:34:28.308+05:30 INFO 21380 --- [main] s.b.a.l.ConditionEvaluationReportLogger : 
```

Error starting ApplicationContext. To display the condition evaluation report re-run your application with 'debug' enabled.

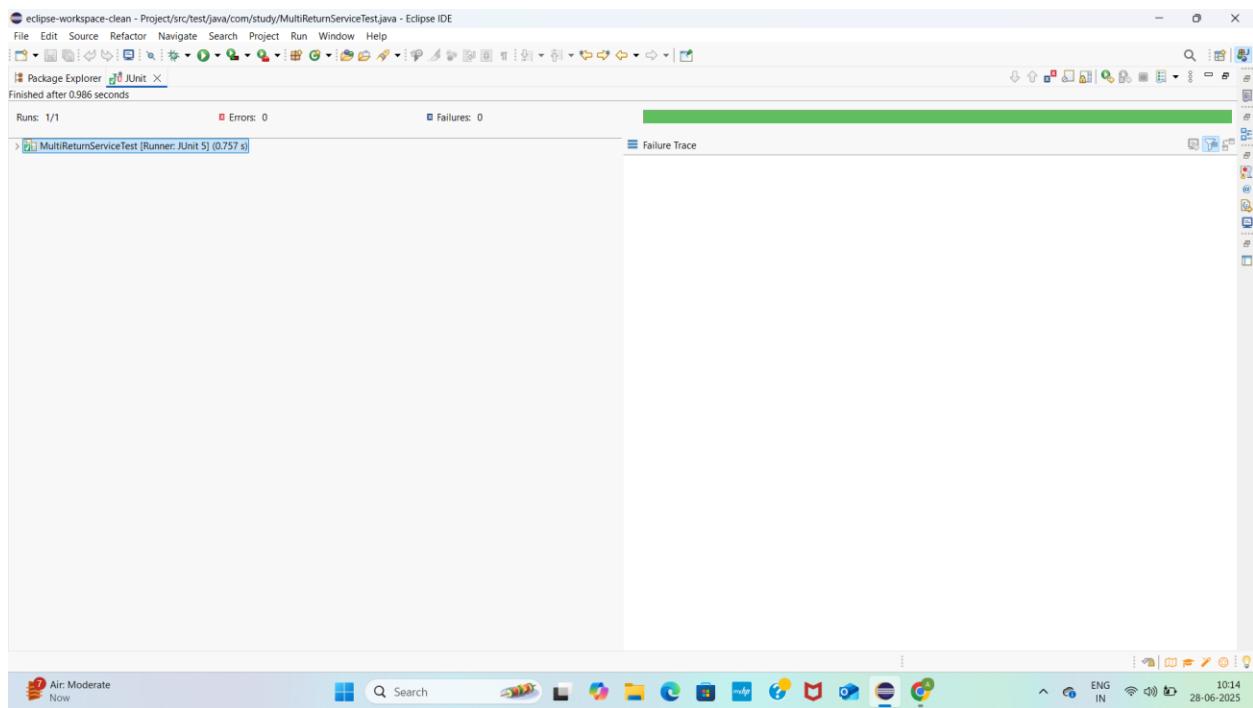
```
2025-06-28T14:34:28.322+05:30 ERROR 21380 --- [main] o.s.b.d.LoggingFailureAnalysisReporter : 
```

APPLICATION FAILED TO START

Description:
Web server failed to start. Port 8080 was already in use.
Action:
Identify and stop the process that's listening on port 8080 or configure this application to listen on another port.

Hot days ahead
36°C

14:34
ENG IN
28-06-2025



2.User.java

```
package com.study;

public class User {
private Long id;
private String name;

public User() {}

public User(Long id, String name) {
this.id = id;
this.name = name;
}

public Long getId() {
return id;
}

public void setId(Long id) {
this.id = id;
}

public String getName() {
```

```
    return name;
}

public void setName(String name) {
    this.name = name;
}
}
```

UserRepository.java

```
package com.study;

import java.util.Optional;

public interface UserRepository {
Optional<User> findById(Long id);
}
```

UserService.java

```
package com.study;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class UserService {

@Autowired
private UserRepository userRepository;

public User getUserById(Long id) {
return userRepository.findById(id).orElse(null);
}
}
```

UserServiceTest.java

```
package com.study;

import static org.junit.jupiter.api.Assertions.*;
import static org.mockito.Mockito.*;

import java.util.Optional;
```

```
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.mockito.InjectMocks;
import org.mockito.Mock;
import org.mockito.MockitoAnnotations;

public class UserServiceTest {

    @Mock
    private UserRepository userRepository;

    @InjectMocks
    private UserService userService;

    @BeforeEach
    public void setUp() {
        MockitoAnnotations.openMocks(this);
    }

    @Test
    public void test GetUserById_UserFound() {
        // Use constructor, no setters needed
        User mockUser = new User(1L, "Alice");

        when(userRepository.findById(1L)).thenReturn(Optional.of(mockUser));

        User result = userService.getUserById(1L);

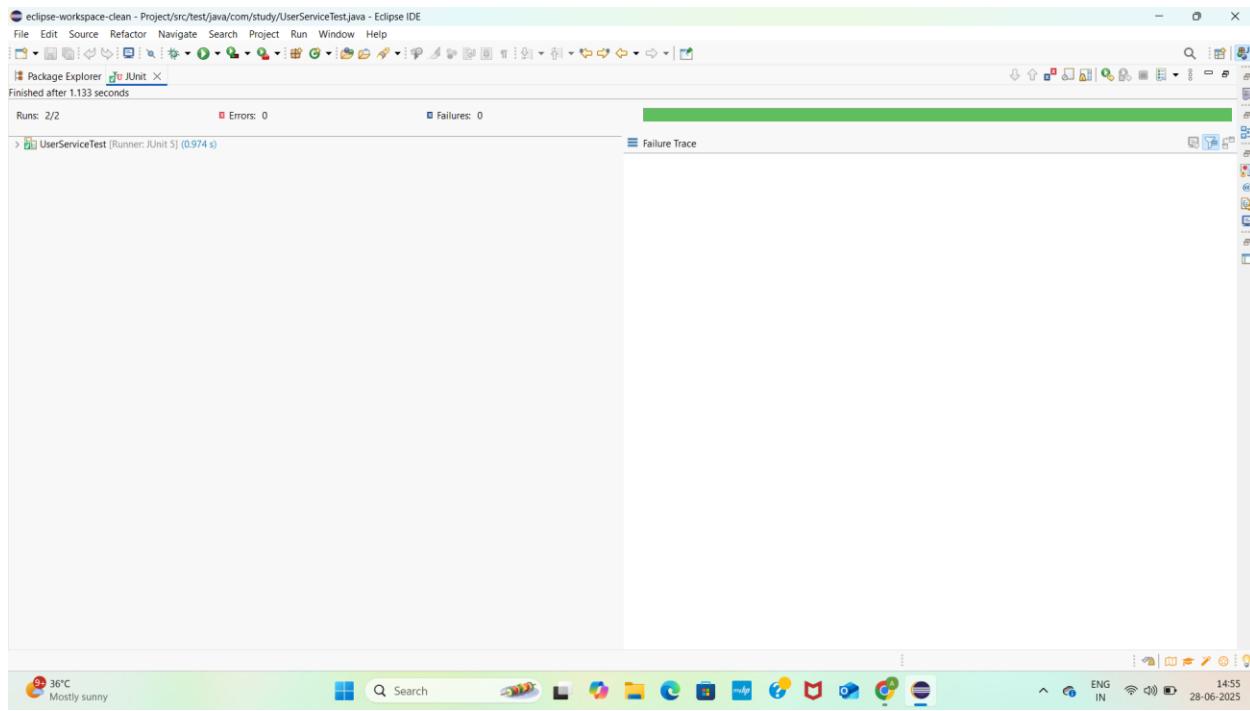
        assertNotNull(result);
        assertEquals("Alice", result.getName());
        assertEquals(1L, result.getId());
    }

    @Test
    public void test GetUserById_UserNotFound() {
        when(userRepository.findById(99L)).thenReturn(Optional.empty());

        User result = userService.getUserById(99L);

        assertNull(result);
    }
}
```

OUTPUTSCREENSHOT



3.

User.java

```
package com.study;

public class User {
    private Long id;
    private String name;

    public User() {}
    public User(Long id, String name) {
        this.id = id;
        this.name = name;
    }

    public Long getId() { return id; }
    public void setId(Long id) { this.id = id; }

    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
}
```

UserService.java

```
package com.study;
```

```
public interface UserService {  
    User getUserById(Long id);  
}
```

UserController.java

```
package com.study;  
  
public class UserController {  
  
    private final UserService userService;  
  
    public UserController(UserService userService) {  
        this.userService = userService;  
    }  
  
    public User getUserById(Long id) {  
        return userService.getUserById(id);  
    }  
}
```

UserControllerTest.java

```
package com.study;  
  
import org.junit.jupiter.api.Test;  
import static org.mockito.Mockito.*;  
import static org.junit.jupiter.api.Assertions.*;  
  
public class UserControllerTest {  
  
    @Test  
    public void test GetUserById() {  
        // Arrange  
        UserService mockService = mock(UserService.class);  
        UserController controller = new UserController(mockService);  
        User mockUser = new User(1L, "John Doe");  
  
        when(mockService.getUserById(1L)).thenReturn(mockUser);  
  
        // Act  
        User result = controller.getUserById(1L);  
    }  
}
```

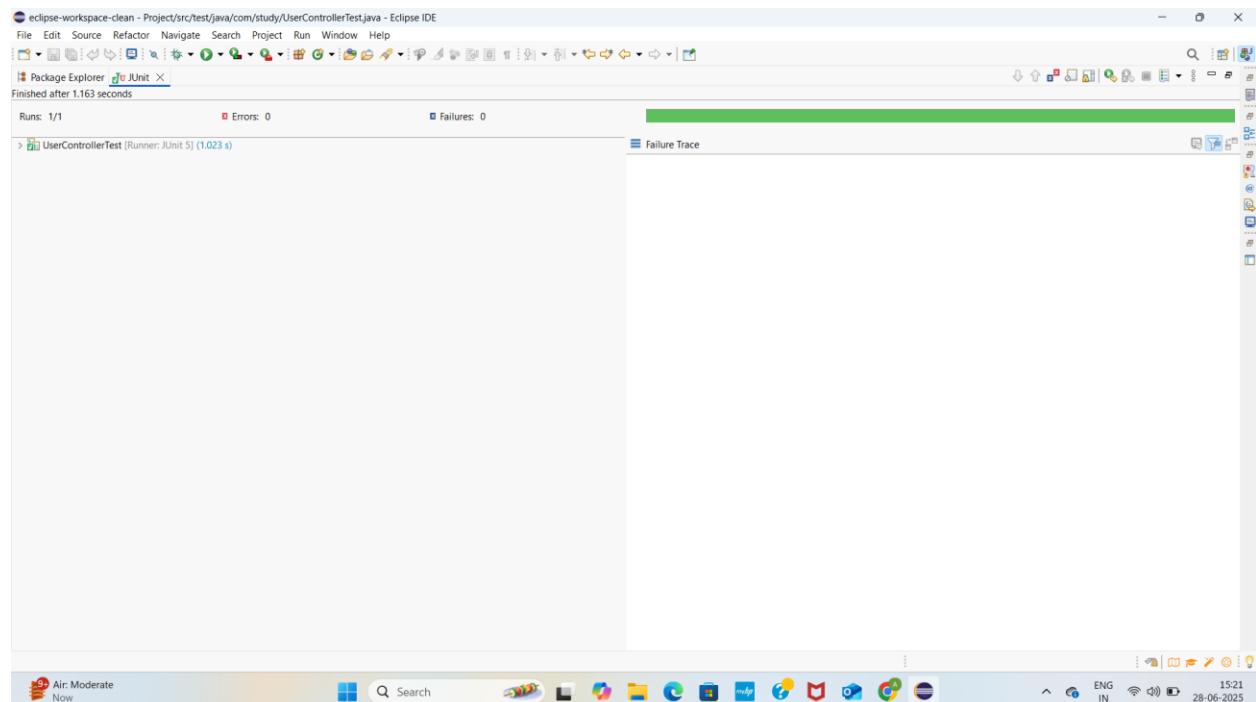
```

// Assert
assertNotNull(result);
assertEquals(1L, result.getId());
assertEquals("John Doe", result.getName());

// Verify interaction
verify(mockService).getUserById(1L);
}
}

```

OUTPUTSCREENSHOT



Logging using SLF4J

1. LoggingExample.java

```

package com.study;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class LoggingExample {

    private static final Logger logger = LoggerFactory.getLogger(LoggingExample.class);

    public static void main(String[] args) {
        logger.error("This is an error message");
    }
}

```

```
Logger.warn("This is a warning message");
Logger.info("This is an info message");
}

}

Pom.xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<groupId>com.study</groupId>
<artifactId>spring-logging-test</artifactId>
<version>1.0-SNAPSHOT</version>

<properties>
<java.version>1.8</java.version>
<junit.version>5.7.0</junit.version>
<mockito.version>3.6.0</mockito.version>
<spring.version>5.3.3</spring.version>
</properties>

<dependencies>

<!--  SLF4J Logging -->
<dependency>
<groupId>org.slf4j</groupId>
<artifactId>slf4j-api</artifactId>
<version>1.7.30</version>
</dependency>

<!--  Logback (SLF4J implementation) -->
<dependency>
<groupId>ch.qos.logback</groupId>
<artifactId>logback-classic</artifactId>
<version>1.2.3</version>
</dependency>

<!--  Spring Core -->
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-context</artifactId>
<version>${spring.version}</version>
</dependency>
```

```
<!-- ☑ JUnit 5 (Jupiter) -->
<dependency>
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter-api</artifactId>
<version>${junit.version}</version>
<scope>test</scope>
</dependency>
<dependency>
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter-engine</artifactId>
<version>${junit.version}</version>
<scope>test</scope>
</dependency>

<!-- ☑ Mockito Core -->
<dependency>
<groupId>org.mockito</groupId>
<artifactId>mockito-core</artifactId>
<version>${mockito.version}</version>
<scope>test</scope>
</dependency>

<!-- ☑ Mockito + JUnit 5 integration -->
<dependency>
<groupId>org.mockito</groupId>
<artifactId>mockito-junit-jupiter</artifactId>
<version>${mockito.version}</version>
<scope>test</scope>
</dependency>

<!-- ☑ Spring Test Support -->
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-test</artifactId>
<version>${spring.version}</version>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<!-- Compiler plugin -->
<plugin>
<groupId>org.apache.maven.plugins</groupId>
```

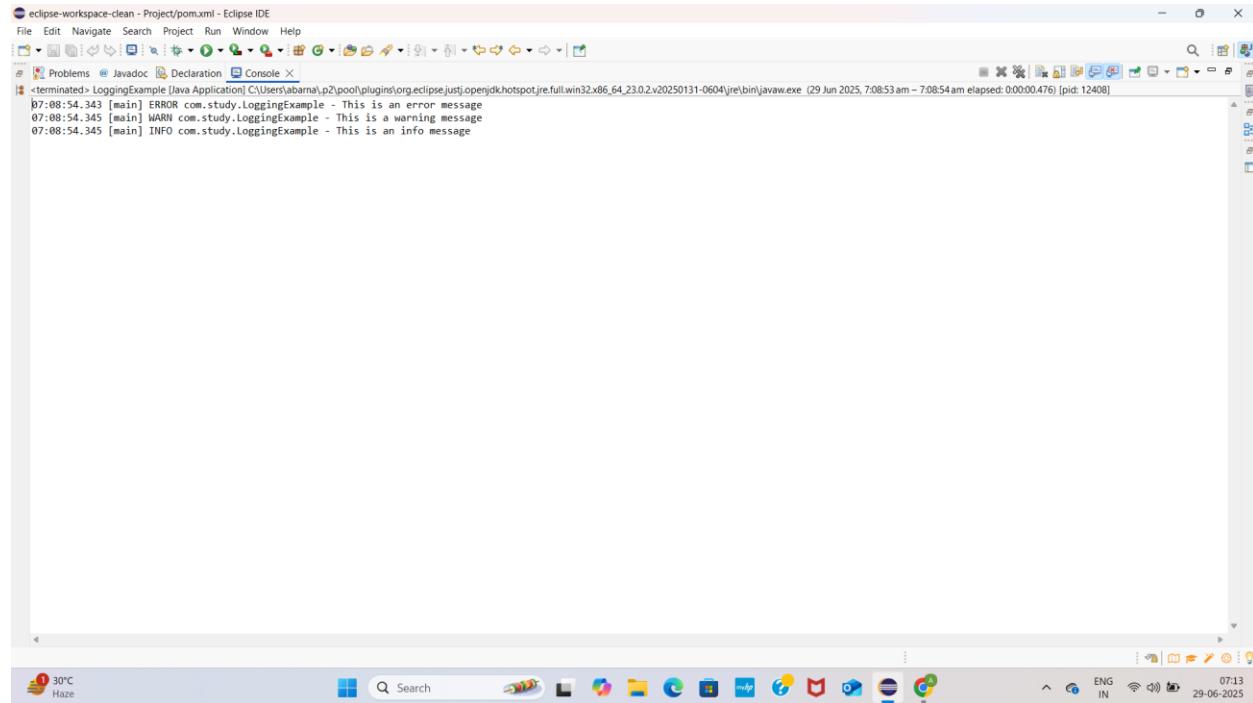
```

<artifactId>maven-compiler-plugin</artifactId>
<version>3.8.1</version>
<configuration>
<source>${java.version}</source>
<target>${java.version}</target>
</configuration>
</plugin>

<!-- Surefire for running JUnit 5 -->
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>2.22.2</version>
<configuration>
<includes>
<include>**/*Test.java</include>
</includes>
</configuration>
</plugin>
</plugins>
</build>
</project>

```

OUTPUT SCREENSHOT



2.

ParameterizedLoggingExample.java

```

package com.study;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class ParameterizedLoggingExample {

    private static final Logger Logger =
LoggerFactory.getLogger(ParameterizedLoggingExample.class);

    public static void main(String[] args) {
String username = "Abarna";
int loginAttempts = 3;

Logger.info("User {} has attempted to login {} times", username, loginAttempts);
Logger.warn("User {} may be locked out after {} failed attempts", username,
loginAttempts);
Logger.error("Login failed for user {} after {} attempts", username, loginAttempts);

// Logging with one parameter
Logger.debug("Current user is {}", username);

// Logging without parameters
Logger.info("Logging completed.");
}

}

Pom.xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<groupId>com.study</groupId>
<artifactId>spring-logging-test</artifactId>
<version>1.0-SNAPSHOT</version>

<properties>
<java.version>1.8</java.version>
<junit.version>5.7.0</junit.version>
<mockito.version>3.6.0</mockito.version>
<spring.version>5.3.3</spring.version>
</properties>

```

```
<dependencies>

    <!-- ☑ SLF4J Logging -->
    <dependency>
        <groupId>org.slf4j</groupId>
        <artifactId>slf4j-api</artifactId>
        <version>1.7.30</version>
    </dependency>

    <!-- ☑ Logback (SLF4J implementation) -->
    <dependency>
        <groupId>ch.qos.logback</groupId>
        <artifactId>logback-classic</artifactId>
        <version>1.2.3</version>
    </dependency>

    <!-- ☑ Spring Core -->
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
        <version>${spring.version}</version>
    </dependency>

    <!-- ☑ JUnit 5 (Jupiter) -->
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter-api</artifactId>
        <version>${junit.version}</version>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter-engine</artifactId>
        <version>${junit.version}</version>
        <scope>test</scope>
    </dependency>

    <!-- ☑ Mockito Core -->
    <dependency>
        <groupId>org.mockito</groupId>
        <artifactId>mockito-core</artifactId>
        <version>${mockito.version}</version>
        <scope>test</scope>
    </dependency>
```

```

<!-- ☑ Mockito + JUnit 5 integration -->
<dependency>
<groupId>org.mockito</groupId>
<artifactId>mockito-junit-jupiter</artifactId>
<version>${mockito.version}</version>
<scope>test</scope>
</dependency>

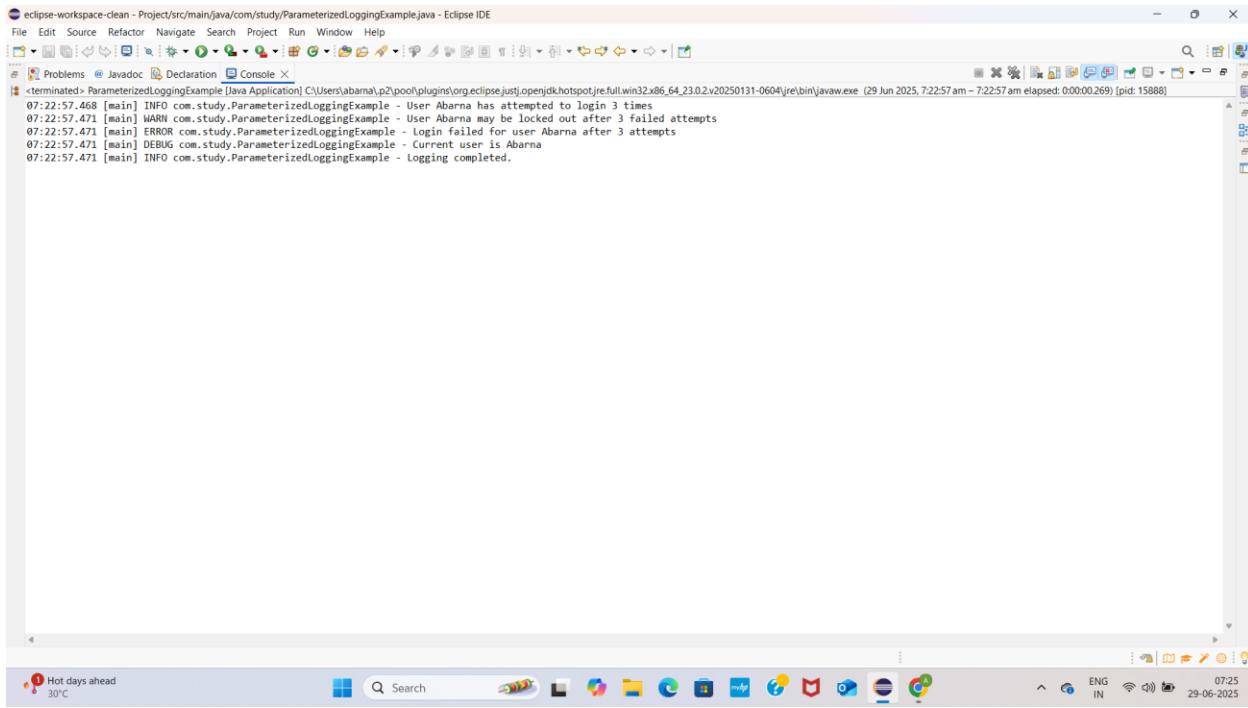
<!-- ☑ Spring Test Support -->
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-test</artifactId>
<version>${spring.version}</version>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<!-- Compiler plugin -->
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-compiler-plugin</artifactId>
<version>3.8.1</version>
<configuration>
<source>${java.version}</source>
<target>${java.version}</target>
</configuration>
</plugin>

<!-- Surefire for running JUnit 5 -->
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>2.22.2</version>
<configuration>
<includes>
<include>**/*Test.java</include>
</includes>
</configuration>
</plugin>
</plugins>
</build>
</project>

```

OUTPUTSCREENSHOT



3.

MultiAppenderExample.java

```

package com.study;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class MultiAppenderExample {

    private static final Logger logger =
        LoggerFactory.getLogger(MultiAppenderExample.class);

    public static void main(String[] args) {
        logger.info("Starting the multi-appender logging example...");
        logger.debug("This is a debug message.");
        logger.warn("This is a warning message.");
        logger.error("This is an error message.");
        logger.info("Finished logging.");
    }
}

```

Logback.xml

<configuration>

<!-- Console Appender -->

```

<appender name="console" class="ch.qos.logback.core.ConsoleAppender">
<encoder>
<pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>
</encoder>
</appender>

<!-- File Appender -->
<appender name="file" class="ch.qos.logback.core.FileAppender">
<file>app.log</file> <!-- File will be created in project root -->
<append>true</append>
<encoder>
<pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>
</encoder>
</appender>

<!-- Root Logger Config -->
<root level="debug">
<appender-ref ref="console" />
<appender-ref ref="file" />
</root>

</configuration>

```

Pom.xml

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<groupId>com.study</groupId>
<artifactId>spring-logging-test</artifactId>
<version>1.0-SNAPSHOT</version>

<properties>
<java.version>1.8</java.version>
<junit.version>5.7.0</junit.version>
<mockito.version>3.6.0</mockito.version>
<spring.version>5.3.3</spring.version>
</properties>

<dependencies>

<!-- ☑ SLF4J Logging -->
<dependency>

```

```
<groupId>org.slf4j</groupId>
<artifactId>slf4j-api</artifactId>
<version>1.7.30</version>
</dependency>

<!-- ☑ Logback (SLF4J implementation) -->
<dependency>
<groupId>ch.qos.logback</groupId>
<artifactId>logback-classic</artifactId>
<version>1.2.3</version>
</dependency>

<!-- ☑ Spring Core -->
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-context</artifactId>
<version>${spring.version}</version>
</dependency>

<!-- ☑ JUnit 5 (Jupiter) -->
<dependency>
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter-api</artifactId>
<version>${junit.version}</version>
<scope>test</scope>
</dependency>
<dependency>
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter-engine</artifactId>
<version>${junit.version}</version>
<scope>test</scope>
</dependency>

<!-- ☑ Mockito Core -->
<dependency>
<groupId>org.mockito</groupId>
<artifactId>mockito-core</artifactId>
<version>${mockito.version}</version>
<scope>test</scope>
</dependency>

<!-- ☑ Mockito + JUnit 5 integration -->
<dependency>
<groupId>org.mockito</groupId>
<artifactId>mockito-junit-jupiter</artifactId>
```

```

<version>${mockito.version}</version>
<scope>test</scope>
</dependency>

<!-- ☑ Spring Test Support -->
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-test</artifactId>
<version>${spring.version}</version>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<!-- Compiler plugin -->
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-compiler-plugin</artifactId>
<version>3.8.1</version>
<configuration>
<source>${java.version}</source>
<target>${java.version}</target>
</configuration>
</plugin>

<!-- Surefire for running JUnit 5 -->
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>2.22.2</version>
<configuration>
<includes>
<include>**/*Test.java</include>
</includes>
</configuration>
</plugin>
</plugins>
</build>
</project>

```

OUTPUTSCREENSHOT

