VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology and Engineering

SOFTWARE TESTING – ITE2004

WINTER SEMESTER 2022-23

PROJECT REPORT
on

DIGILOCKER SYSTEM

By

JAISHREE - 20BIT0297
KALAMEGAM V - 20BIT0302
PRIYA N - 20BIT0305
PRIYADARSHINI.R - 20BIT0307
ABRETHA BEGAM.A - 20BIT0309

submitted to

Dr.IYAPPARAJA M
Associate Professor Grade 1
School of Information Technology and Engineering

ABSTRACT :

One should always have a valid identity proof with them at all times. Having our Aadhaar card or PAN card or any other important document with us is a must. But carrying them around in our wallet or bag could be risky since it could be misplaced. And even worse, if it gets in the hands of a malicious person, our personal information could be at risk. Therefore, in the present world where everything is being digitized, it is time we do the same with all our important documents.

Digi locker is an Indian government-approved website which lets us store digital copies of all our important documents. The documents are issued to us by registered issuers and is therefore considered a legally valid soft copy. That is, it gets the same treatment as the original documents. So, whenever we need to present any document, we can simply use this digital copy. The main advantage of using Digi locker is that we are now able to access a whole range of official certificates/documents (Aadhar card, PAN card, driving license, birth certificate, class X and class XII mark sheets to name a few) from anywhere and at anytime.

Considering all the points we just mentioned, it makes us ponder over one question. Is this website secure? Does this website do what it actually is supposed to do? It becomes an inevitable task for us to test this website in order to get answers to these questions. At the end of the day, all our personal information is contained here and it could be compromised if the website has some defect or is not working accordingly to meet its requirements. **Performing software testing helps us identify if there are any problems or errors** in the system. The main need for software testing in this case is to ensure **security**.

INTRODUCTION:

Currently, in India, almost all of the government issued documents are in physical form across the country. This means every time a resident need to share the document with an agency to avail any service, an attested photo copy either in physical form or on scanned form is shared. Use of physical copies of document creates huge overhead in terms of manual verification, paper storage, manual audits, etc. incurring high cost and inconvenience. This creates problem for various agencies to verify the authenticity of these documents, thus, creating loopholes for usage of fake documents/certificates. Due to the nature of these documents not having a strong identity attached to it, anyone with same name can indeed misuse someone else's document.

Targeted at the idea of paperless governance, Digi Locker is a platform for issuance and verification of documents & certificates in a digital way, thus eliminating the use of physical documents.

The purpose of this document is to present a detailed description of the Digi Locker system. It will explain in purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document it intended for both the stakeholders and the developers of the system.

The project is intended to create a website where citizens can download their all type of government approved certificates that's connected to our Aadhar number. **Citizen:** A legally recognized subject or national of a state or commonwealth, either native or naturalized.

LITERATURE SURVEY:

S.No	Author,Year	Title	Methodology	Pros	Cons
1.	Urko Rueda Molina,Fitsum Kifetew, Annibale Panichella. 2018	Java Unit Testing Tool Competition - Sixth Round	Public contest repository,JUnit tools setup,CUTs, Execution frame,Testgeneration, Metrics computation,Combined analyses,Time budgets.	Generate the test case in Less time.	large budgets would likely result in higher overall performance for the blended check suites over the individual device outcome and the human-developed tests.
2.	Jean Petric,Tracy Hall ,David Bowes 2018	How Effectively Is Defective Code Actually Tested? An Analysis of JUnit Tests in Seven Open Source Systems	First extract the defects from seven open source projects the usage of the SZZ algorithm. Suit these defects with JUnit checks to pick out the share of defects that had been covered by JUnit tests. Additionally do the equal for non-defective code. Then usePrincipal Component Analysis and laptop gaining knowledge of to investigate the traits of JUnit assessments that had been profitable in identifying defects.	show that the number of methods touched by a JUnit test is strongly related to that test uncovering a defect	results show that a large number of defective methods are not covered by JUnit tests.

3.	D. Ma'ayan 2018	The quality of junit tests: an empirical study report	Introduce a static evaluation approach for analysing of large corpuses of code. For every corpus, traverse over all projects within it, and in every undertaking traverse over all its Java files. Each Java file is translated into an AST, and at some stage in a traversal over the AST, pattern,are, recognized. Finally, all the facts is saved in a CSV file for a similarly analysis.	These early results might demonstrae the importance of automation tools and the need for refactoring techniques for unit tests.	Here not find correlations between our results to traditional testing metrics such as code coverage by tests.
4.	R. MUKHERJEE AND K. S. PATNAIK 2019	Prioritizing JUnit Test Cases Without Coverage Information: An Optimization Heuristics Based Approach	Threads of validity, Java XML parser.	it appears clearly that application of optimization heuristics produce great benefifit for ordering JUnit test cases at test method level.	Re-execution of all scheduled take a look at instances is not possible because of constraint trying out time window. We hope the positive results of our find out about will make utilization of optimization algorithms as a promising choice for prioritizing JUnit take a look at cases at take a look at approach level.

5.	Venkatesan, Praveen Kumar; Gade Rozario, Rikhil; Fiaidhi, Jinan,2020	Junit frame work for unit testing.	Mongodb.	This improves programmer effificiencyand system code reliability, which in effect decreases programmer frustration and debug time. This enhances the developer's effificiency.	The software may be used in as a whole or in components. If a product is to be appropriate for usage, each test must be passed.
6.	Xavier Devroey,Sebasti ano Panichella, Alessio Gambi, 2020	Java Unit Testing Tool Competition - Eighth Round	Similar to previous edition Public contest repository , Execution environment, Test generation and time budget, Metrics computation, Combined analysis and comparison with manually written tests, Statistical analysis.	According to this paper most of the bugs mentioned in seventh edition are corrected.	Individual tool running on two different hardware may cause side effects.To reduce the impact resource has to be limited
7.	Maxim L. Gromov, Svetlana A. Prokopenko, Natalia V. Shabaldina, Andrey V. Laputenko Tomsk State University, Tomsk, Russia - 2019	Model Based JUnit Testing	Formal models of programs and tests, Mapping a Java class to a formal model, FSMTest2JUnit	Based on this paper , the created tool is used to test telecommuni cation protocols ordinary Java application and students UML Diagram design implementati on	Need to improve programmes FSMTest2JUnit and TFSMTest2\ JUnit to automate the formal model's extraction from the Java class.

8.	Rakshith D C1, Dr. Manjunath A E2 , 2020	A Comprehensive Study on Automation Testing using JUnit	During the configuration step, an object hierarchy for the test case is established, with each item standing in for a different use-case. Programmers or developers implement these subclasses, and each of their methods must be annotated with the phrase "Test" in order to establish the object hierarchy. The "Test" methods are where the domain-andapplication specific code is written.	Open source framework for java development. It is useful to those who lack knowledge about unit testing by generation of automation test case and execution.	Not applicable for customer-driven testing .Lacking technical knowledge it makes difficult to understand IDE errors.
9.	Kifetew, Fitsum ; Devroey, Xavier; Rueda, Urko 2019	Java Unit Testing Tool Competition - Seventh Round	Public contest repository10 , Execution frame, Test generation, Metrics computation, Combined analyses,Time budgets,	The ease with which a particular tool might be installed and used in the contest environment was demonstrated by the fact that the Contest infrastructure is dockerized.	Docker is not available in the cluster environment where we perform all of the studies, so we are unable to use it to complete the experiment. Due to this, keeping the two versions synchronised required more work

10.	Danielle Gonzalez, Suzanne Prentice Mehdi Mirakhorli, 2018	A Fine- rained Approach for Automated Conversionof to JUnit Assertions English	Identify the assert's parameters, convert the expected and actual value parameters to English phrases, and combine these phrases into a single English sentence based on the assert's condition	English summaries of JUnit tests will become more valuable and accurate as a result of a thorough set of heuristics, improving test maintainabilit y and traceability. The procedure was also put into practis e and made available as the AssertConvert tool	need to improve parameter conversion heuristics, add heuristics for assert .
11.	Julian Harty,2021	Security testing using JUnit and Perl scripts	A rigorous and thorough approach to identify and address potential vulnerabilities in software systems then using Perl scripts to automate the testing and analysis of the results.	Comprehensi ve testing, Costeffective, Repeatable and scalable, Integration with existing processes	Limited coverage, False positives/negatives,Techn cal expertise, Tool limitations

12	Jianwei Wu, James Clause,2023	Automated Identification of Uniqueness in JUnit Tests	The methodology has the potential to improve the efficiency of test suite maintenance by identifying redundant test cases and reducing the size of the test suite without sacrificing test coverage.	Improved Efficiency, Increased Test Effectiveness, High Accuracy, Flexibility, Practicality	Theproposed methodology may not be scalable for very large test suites with a high number of test cases. The proposed methodology is specifically designed for JUnit test suites and may not be applicable to other types of test suites or testing frameworks.
13.	Christopher Vendome Maurício Aniche Christoph Treude Marco Aurélio Gerosa,2019	UniVerse: largescale JUnit-test analysis in the wild	Provides a comprehensive approach to analyzing JUnit tests and can be used to gain insights into how developers write tests and how testing practices evolve over time.	The methodology collects multiple metrics and data points about the tests, including their purpose, metrics, evolution, coverage, and statistical analysis. This provides a nuanced view of testing practices and trends.	The methodology is limited to analyzing JUnit tests in Java, and may not be applicable to other programming languages or testing frameworks. The methodology only analyzes the test code and does not consider other factors that may affect testing practices, such as developer experience or project constraints.

14.	Elvys Soares; Marcio Ribeiro; Rohit Gheyi; Guilherme Amaral; Andre Medeiros Santos,2022	Refactoring Test Smells With JUnit 5: Why Should Developers Keep Up-to-Date	Provides a comprehensive approach to identifying and refactoring test smells in JUnit 5 tests, with potential applications for improving the quality and maintainability of test suites.	The refactoring recommendations provided in the study are based on best practices and guidelines for effective test writing, and are supported by empirical evidence. he methodology can be extended to other testing frameworks and programming languages, providing a general approach to identifying and refactoring test smells that can be applied in a range of contexts.	Did not explore the tradeoffs between different refactoring strategies and did not consider the impact of the recommendations on test performance or coverage.
15.	Boni García, Diego Molina Paco, Saavedra, 2019	Selenium-Jupiter: A JUnit 5 extension for Selenium WebDriver	Provides a practical approach to automating web testing using Selenium WebDriver and JUnit 5, with the Selenium-Jupiter extension providing additional functionality and convenience for managing the test environment	The extension provides built-in reporting capabilities, including capturing screenshots and videos of test execution, which can help diagnose failures and issues. The extension can be easily integrated	Web testing automation requires ongoing maintenance and upkeep to keep the test suite up to date and effective, which can be time consuming and resource intensive.

			and analyzing results.	with existing tools and frameworks, including CI/CD pipelines and other testing frameworks, making it a flexible and versatile tool for web testing automation.w	
16.	An Experiment on the Effects of Modularity on Code Modification and Understanding. 2020	Ewan Tempero, Kelly Blincoe, Danielle Lottridge	Modularity has been referred to for decades with variations on the construct and a gap in evidence on the effects on programmers. they conducted an experiment to investigate the causal impact of modularity on software engineering performance. The results indicate that designs with high modularity code enable participants to achieve successful modification compared to designs with low modularity.	modularity on modifiability	programmes are developed according to international curriculum not developed based on outside of an organization

17.	A Big Data Platform for International Academic Conferences Based on Microservice Framework. 2018	Biao Yang , He Liu , Xuanrui Xiong , Shuaiqi Zhu , Amr Tolba and Xingguo Zhang	In methodology phase, the Java checking out framework Junit used to be used to function unit checking out on a function-by-function basis. We used printouts and breakpoint debugging to take a look at the shape of loops and branches inside functions, calls between functions, and facts interactions between layers to make sure that no mistakes appear in the program. Additionally, the proper jogging surroundin	online tutorial convention pr ovider platfor m primarily based on the proposed con vention sugge stion approac h can furnish handy and qui ck convention organisation and participation o fferings for co nvention orga	scholars can't quickly obtain high-quality conference information and broaden their research fields.

18.	On the Distribution of Test Smells i Open urce Android Applications: An Exploratory Study. 2022	Anthony Peruma,Khalid Saeed Almalki , Christian D. Newman, ,Moham ed Wiem Mkaouer ,Ali Ouni	They investigated the diagram of unit assessments by way of analyzing the occurrence of check smells in Android apps and their have an impact on on the overall quality of the apps thru a set of quantitative, comparative and empirical experiments. they accrued facts for these experiments . Information about they separated take a look at from production archives is reacha ble through the mission intern et site .they carried out a two phased strategy that consisted of: data collection and scent detection. In the first phase, they collected datasets from more than one sources, whilst in the 2nd phase, they analyzed the gathered datasets to realize check smells, along side with the project metadata, wanted later for the experiments..	helps developers to build and maintain better quality test cases for Android apps	represents a threat to test file's maintainability
19.	Practitioner Perceptions of Ansible Test Smells. 2023	Yue Zhang,Fan Wu,Akond Rahman	they answer RQ by conducting an online surveywith practitioners,who develop IaC manifests. they first asked practitioners about their experience in developing IaC manifests. Next, they asked how frequently practitioners test IaC	Ansible manifests have relevance amongst practitioners	researchers didn't investigate why test smells have varying practitioner perceptions

			manifests using a Likert scale		
20.	A complete automation of unit testing of Java programs. 2019	Yoonsik Cheon, Myoung Yee Kim, Ashaveena Perumandla	JUnit is a open-source unit testing framework for Java and provides a way to organize test data and perform test execution. In JUnit, one has to write Java code, called a test class, that describes testdata, invokes the methods to be tested, and determines test results.	genetic algorithms to objectoriented programs by addressing the issues of genetic encoding, genetic operations, and fitness of objects.	limited to simple data types

TEST CASE REPORT:

Module 1: Signup

TEST CASE ID	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
T1		Name is required	Name is required	Accepted
T2	Abretha	Success	Success	Accepted
T3		Name is required	Success	Failed
T4		Selected date, month ,year	Selected date, month ,year	Accepted
T5	Mar-2023	Select date	Select date	Accepted
T6	29--2023	Select month	Select month	Accepted
T7	29-mar-	Select year	Select year	Accepted
T8	--2003	Select date, month	Select date, month	Accepted
T9	-mar-	Select date, year	Select date, year	Accepted
T10	23--	Select month, year	Select month, year	Accepted
T11		Select gender	Select gender	Accepted

T12	Male	Success	Success	Accepted
T13		Mobile number is required	Success	Failed
T14	456789	Invalid mobile number	Invalid mobile number	Accepted
T15		Mobile number is required	Mobile number is required	Accepted
T16	78787878	Invalid number	Success	Failed
T17	678	Pin must be 6 digits	Pin must be 6 digits	Accepted
T18	67890123	Pin must be 6 digits	Pin must be 6 digits	Accepted
T19		Security pin is required	Security pin is required	Accepted
T20	12345	Pin must be 6 digits	success	Failed
T21		Email is required	Email is required	Accepted
T22	Priya.com	Invalid email	Invalid email	Accepted
T23	dharshini@	Invalid email	success	Failed
T24		Aadhar number is required	Aadhar number is required	Accepted

T25	12345678	Invalid Aadhar must be 12 digit only	Invalid Aadhar must be 12 digit only	Accepted
T26	12345678	Invalid Aadhar number	Invalid Aadhar number	Accepted

MODULE 2:MARKSHEET DOWNLOAD

TEST CASE ID	**INPUT**	**EXPECTED OUTPUT**	**ACTUAL OUTPUT**	**STATUS**
TC1		Name is required	Name is required	Accepted
TC2	909345	success	success	Accepted
TC3		Roll number is required	Roll number is required	Accepted
TC4	909349	Invalid roll number	Invalid roll number	Accepted
TC5		Year is required	Year is required	Accepted
TC6	2016	success	success	Accepted
TC7	2013	2016-2019 documents only available	2016-2019 documents only available	Accepted
TC8	Regular	Success	Success	Accepted
TC9		Select certificate type	Select certificate type	Accepted

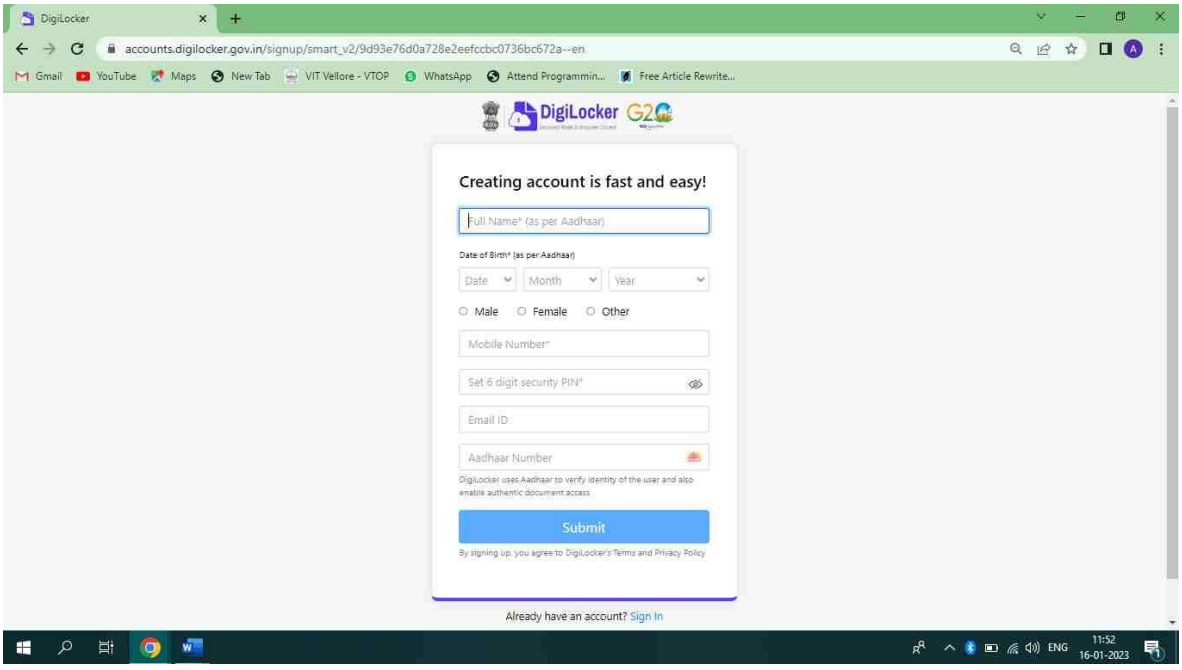
TC10		Select flag	Select flag	Accepted
TC11		Success	Success	Accepted
TC12	JAN	Success	Success	Accepted
TC13	JHK	Invalid month	Invalid month	Accepted
TC14		Month is required	Month is required	Accepted

MODULE:3 PAN CARD DOWNLOAD

TEST CASE ID	**INPUT**	**EXPECTED OUTPUT**	**ACTUAL OUTPUT**	**STATUS**
TC1	123	Name is required	Name is required	Accepted
TC2		Gender is required.	Gender is required.	Accepted
TC3		DOB is required	DOB is required	Accepted
TC4	PDA414	Invalid PAN number	Invalid PAN number	Accepted
TC5		PAN number is required	PAN number is required	Accepted
TC6	PDA414KJII	Invalid PAN number	Invalid PAN number	Accepted

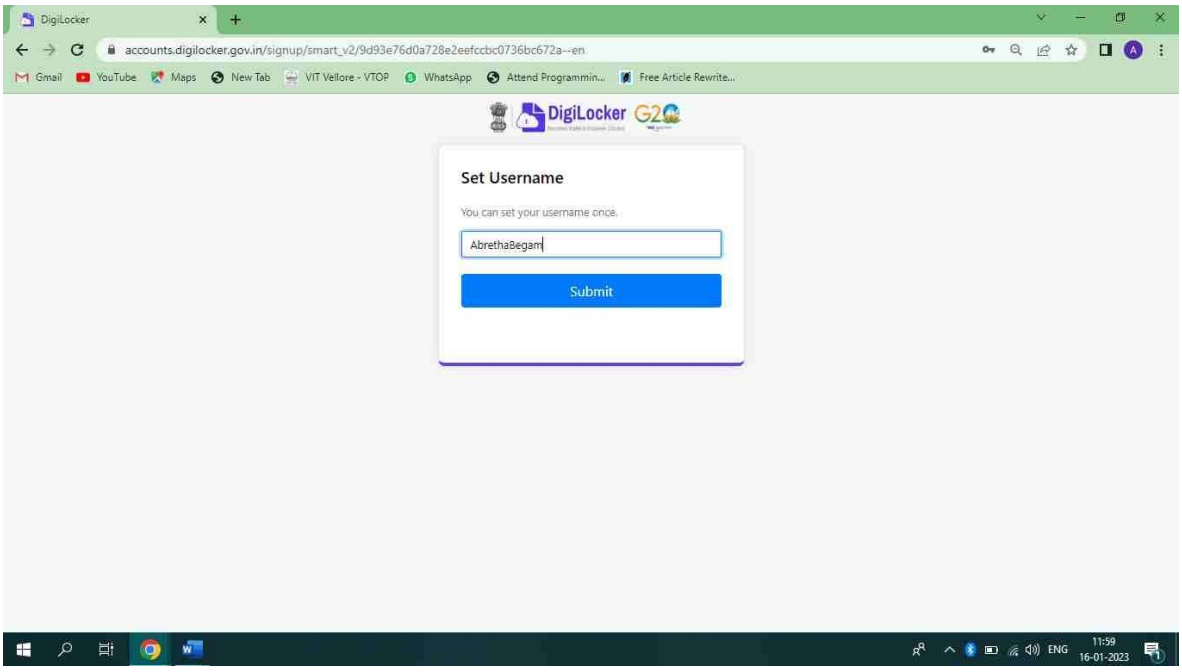
SAMPLE OUTPUT SCREENSHOTS :

By clicking Sign Up :



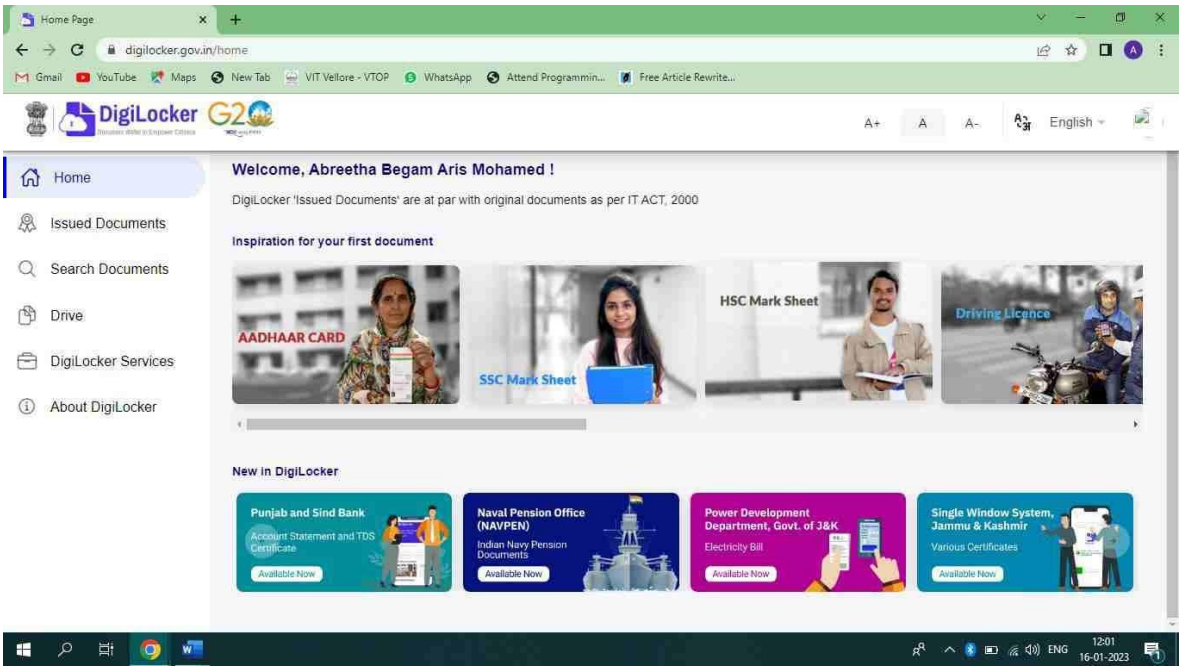
After Clicking Submit Verifying through OTP :

Setting Username:



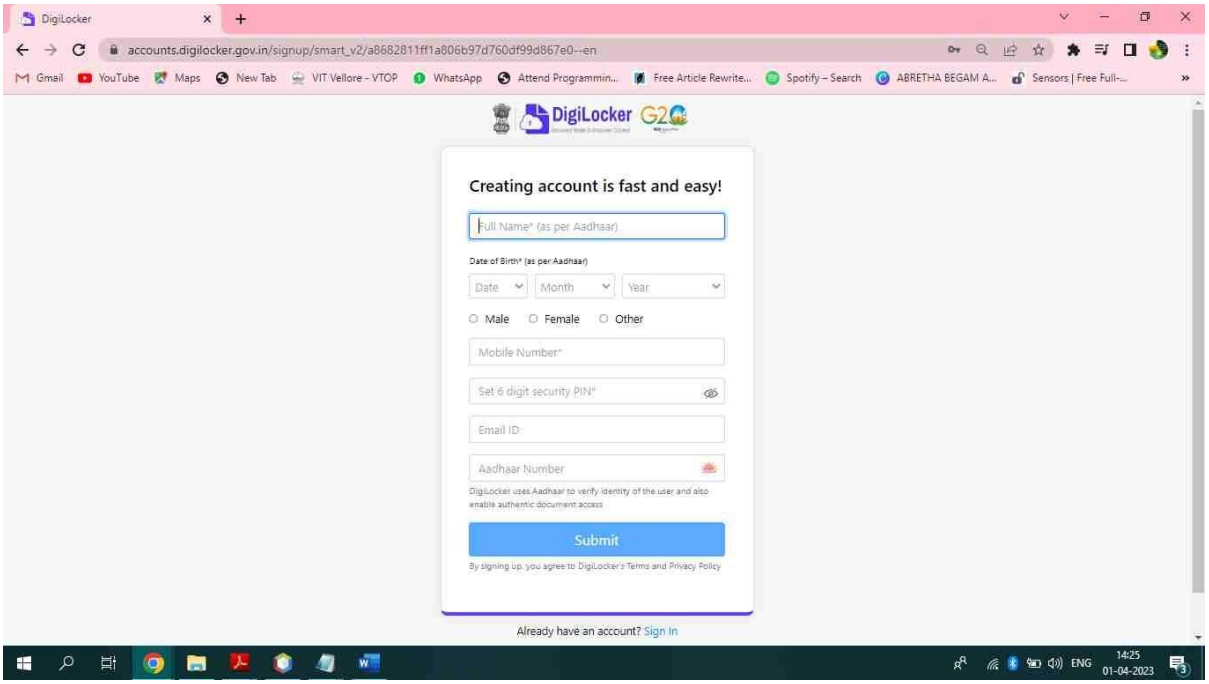
After Sign Up Sign In

Home Page :

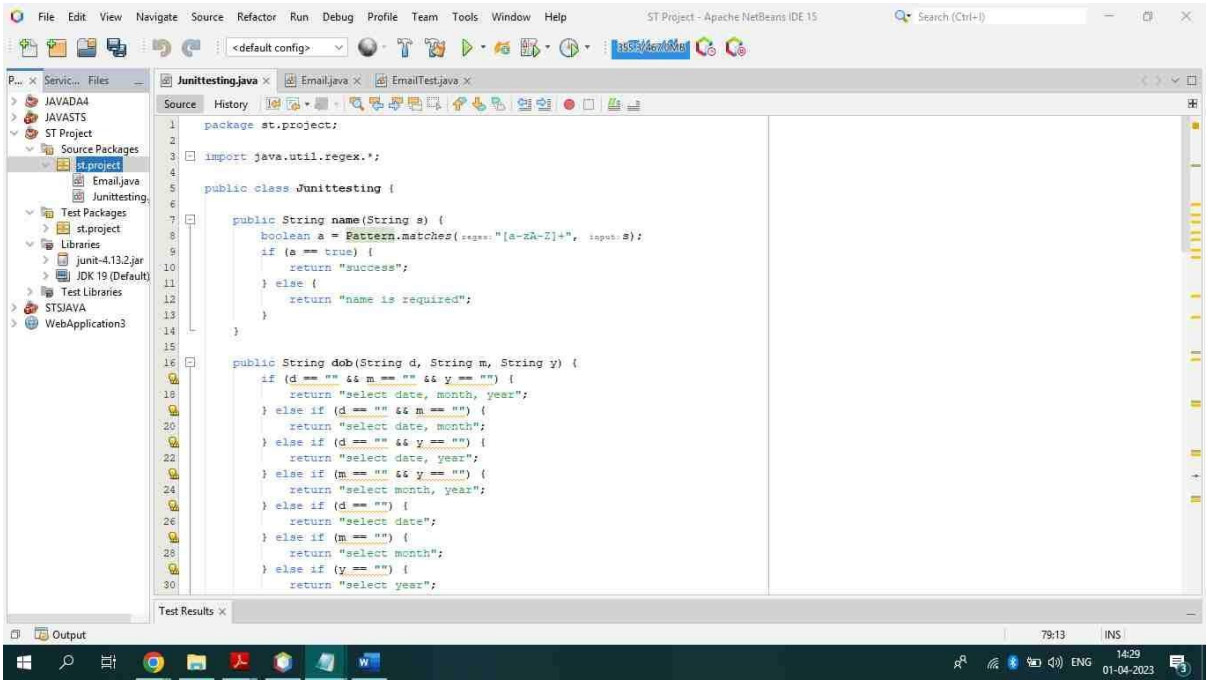


DIGILOCKER SYSTEM:

Module 1: Signup



Junittesting.java

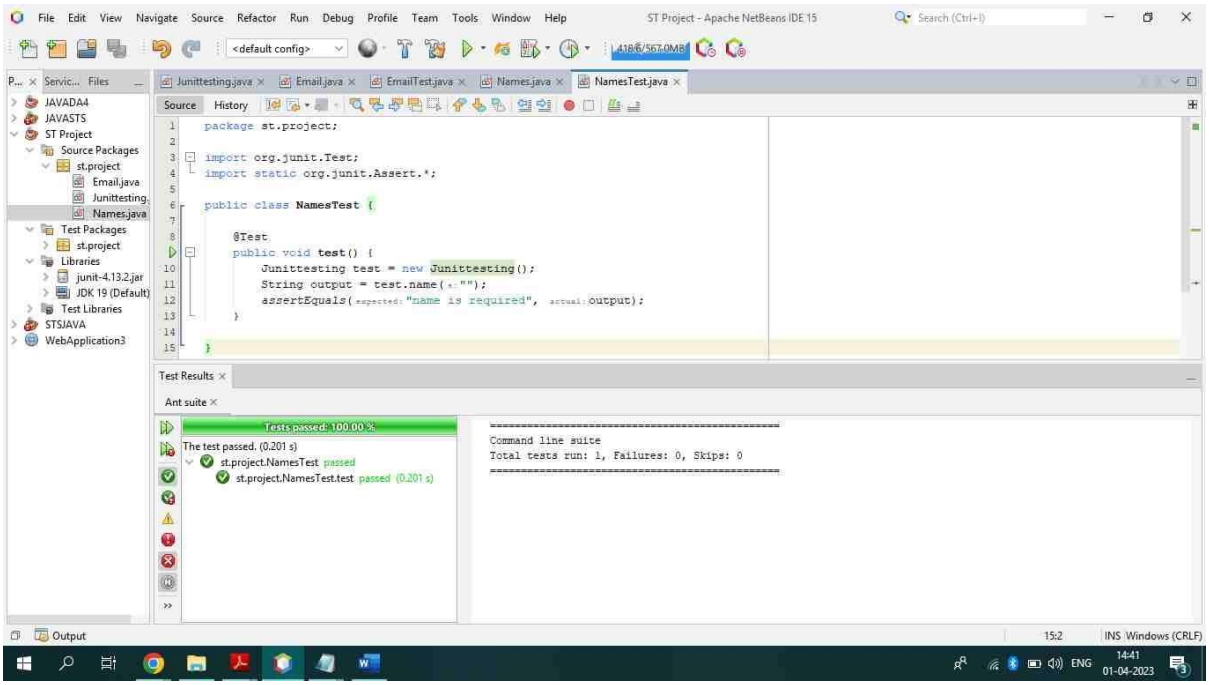


Unit 1: Name

Test case ID: T1

Input:

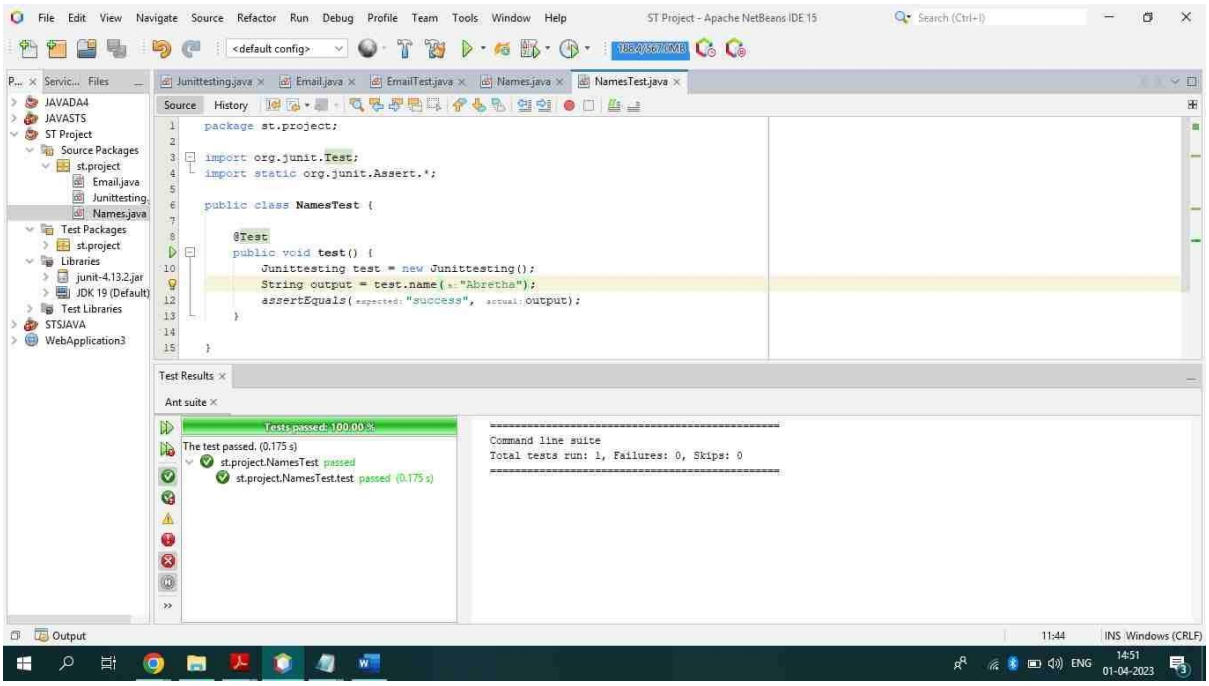
Expected output: “name is required”



Test case ID: T2

Input: Abretha

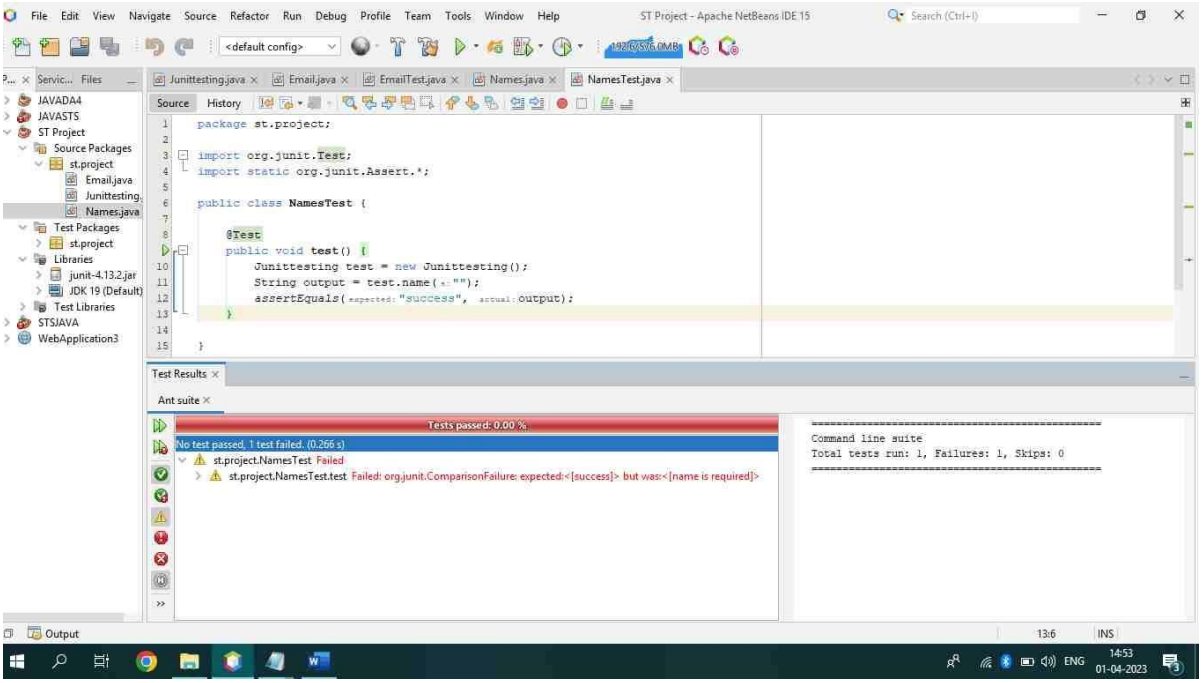
Expected output: “Success”



Test case ID: T3 (Failed Test Case)

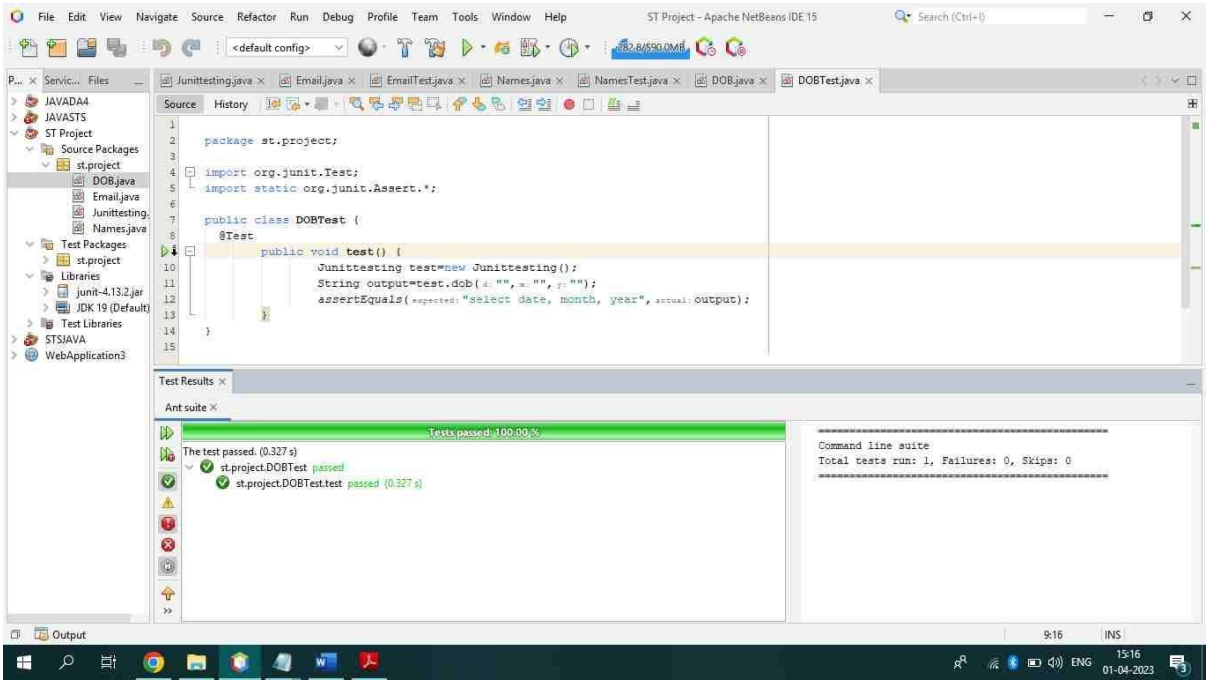
Input:

Expected output: “name is required”

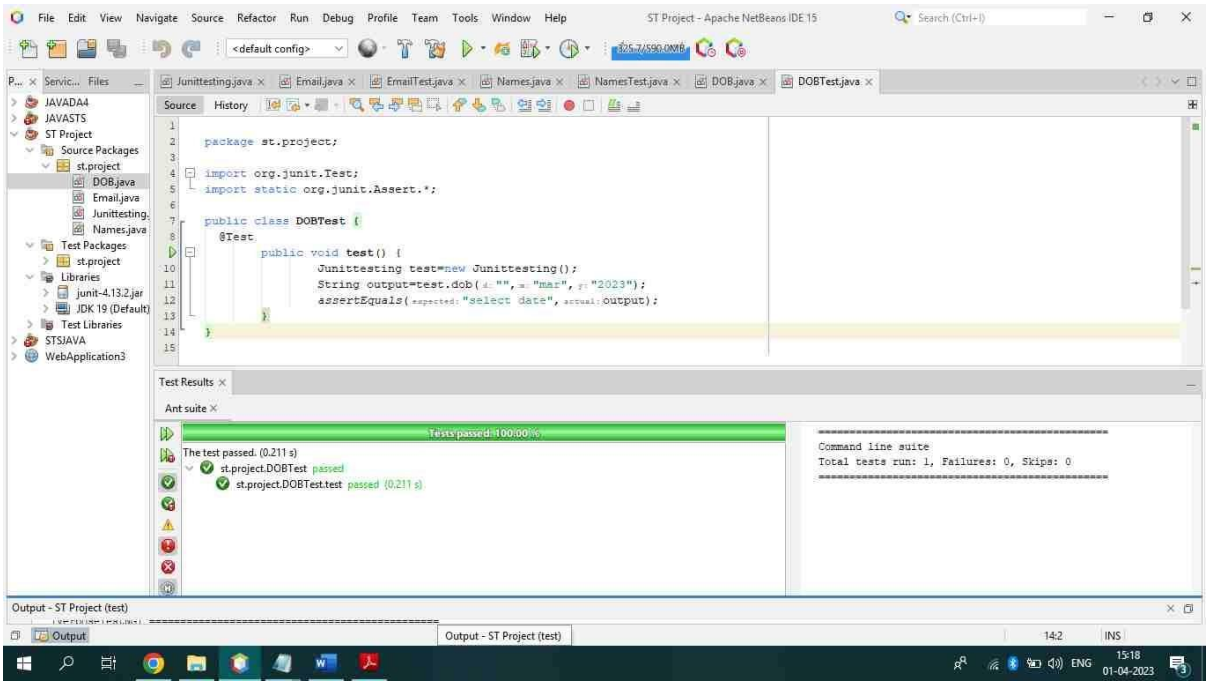


Unit 2: DOB

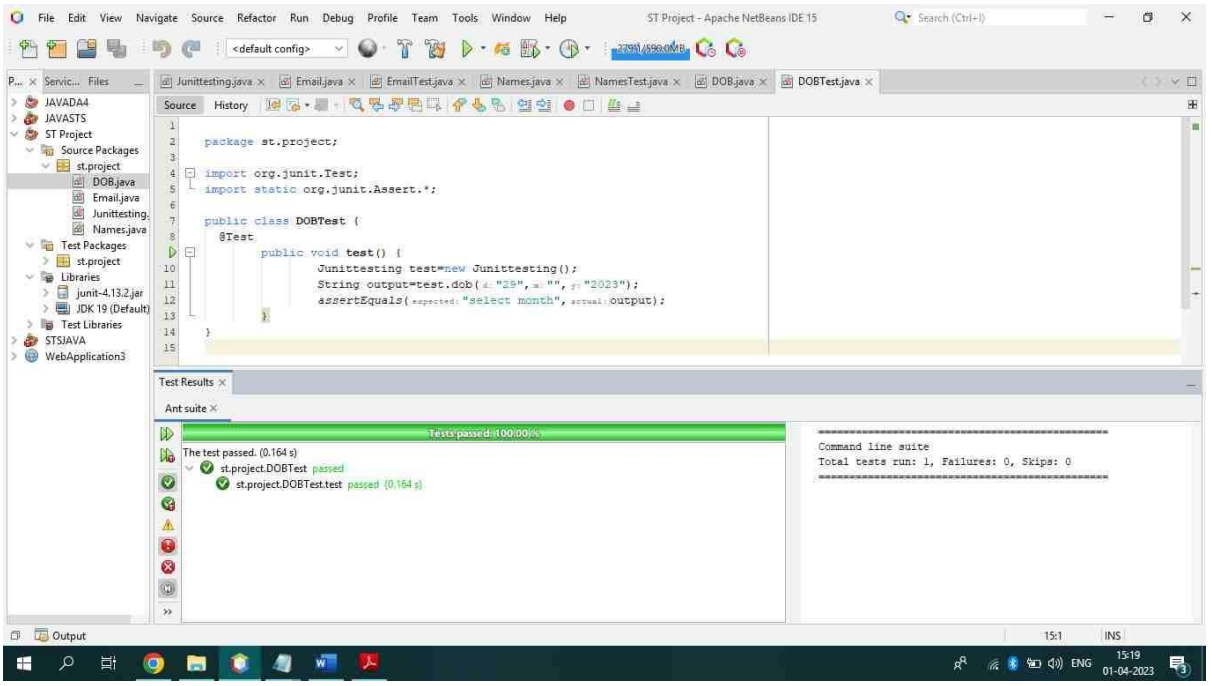
Test case id: T4
Input:
Expected output: select date,month,year



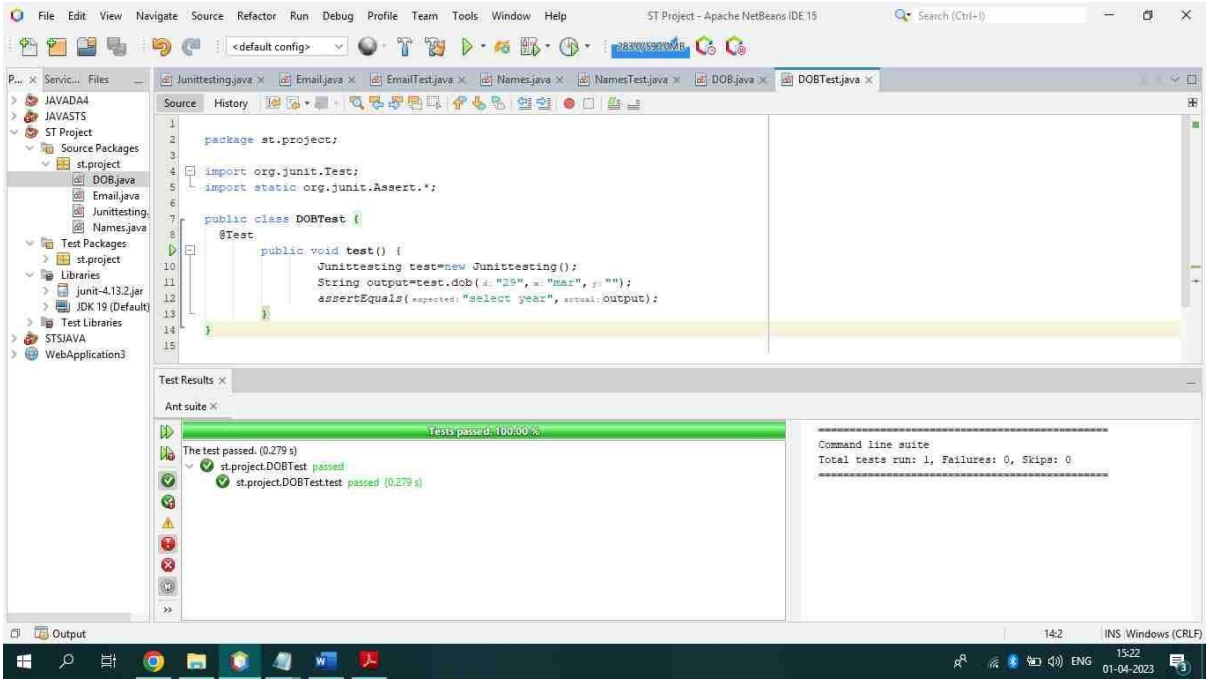
Test case id: T5
Input: -mar-2023
Expected output: select date



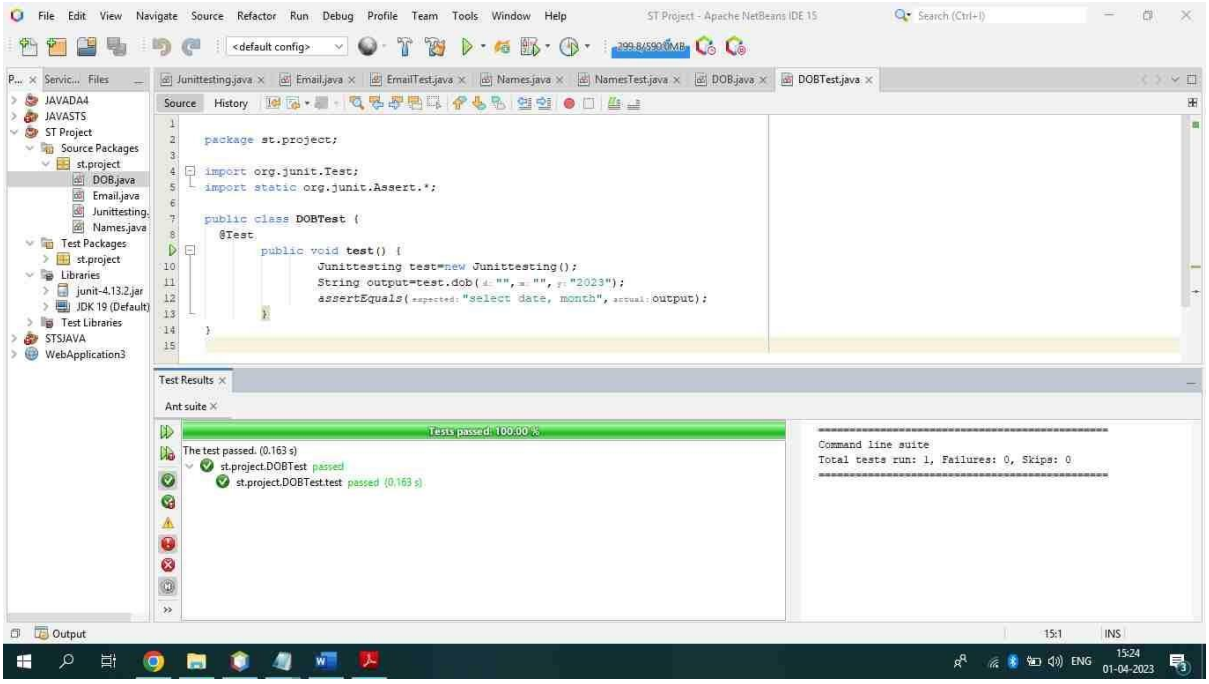
Test case id: T6
Input: 29- -2023
Expected output: select month



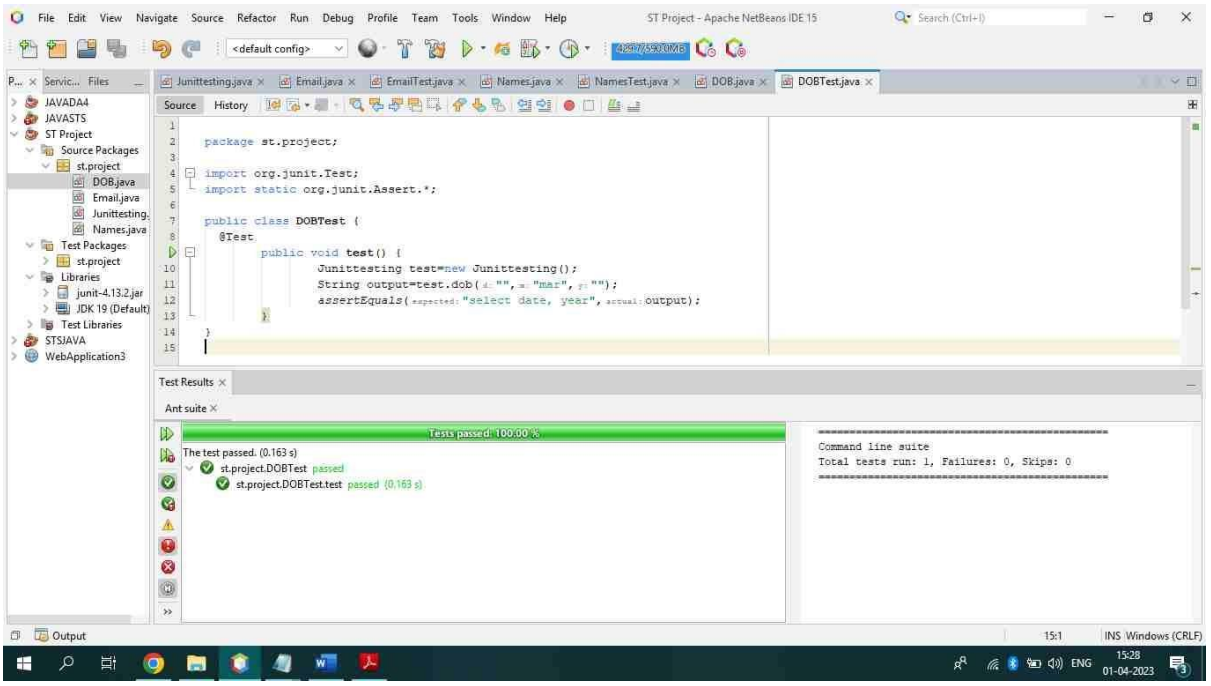
Test case id: T7
Input: 29- mar-
Expected output: select year



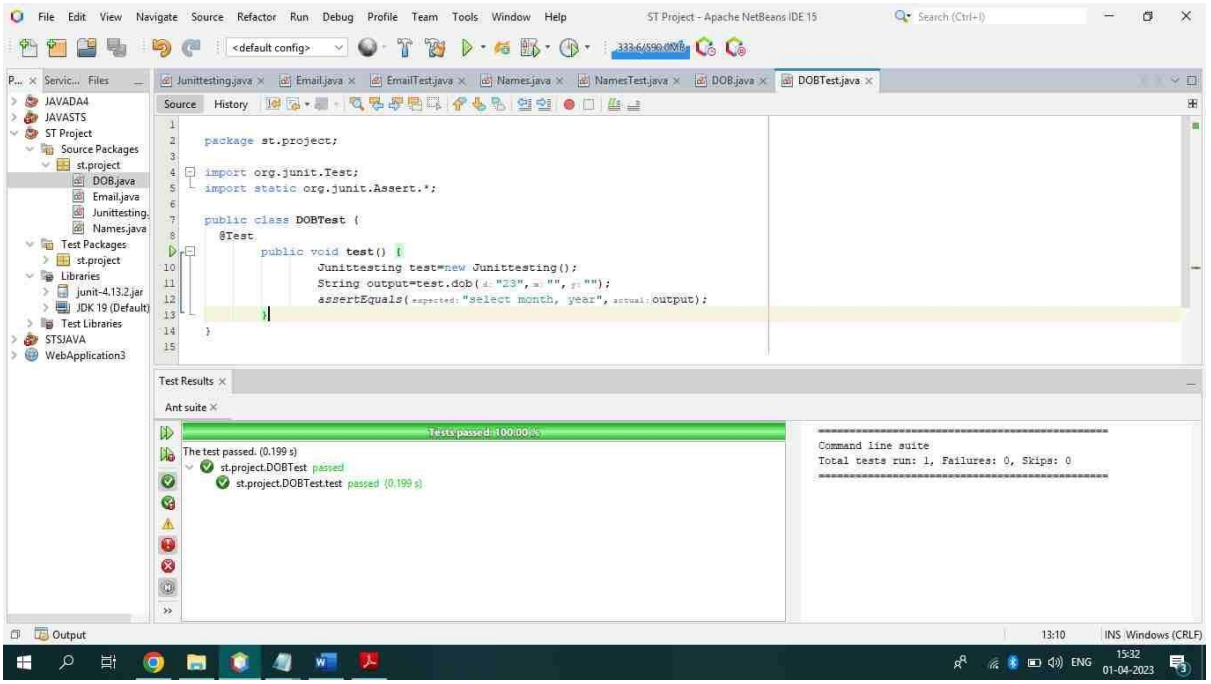
Test case id: T8
Input: - -2003
Expected output: select date, month



Test case id: T9
Input: - mar-
Expected output: select date, year

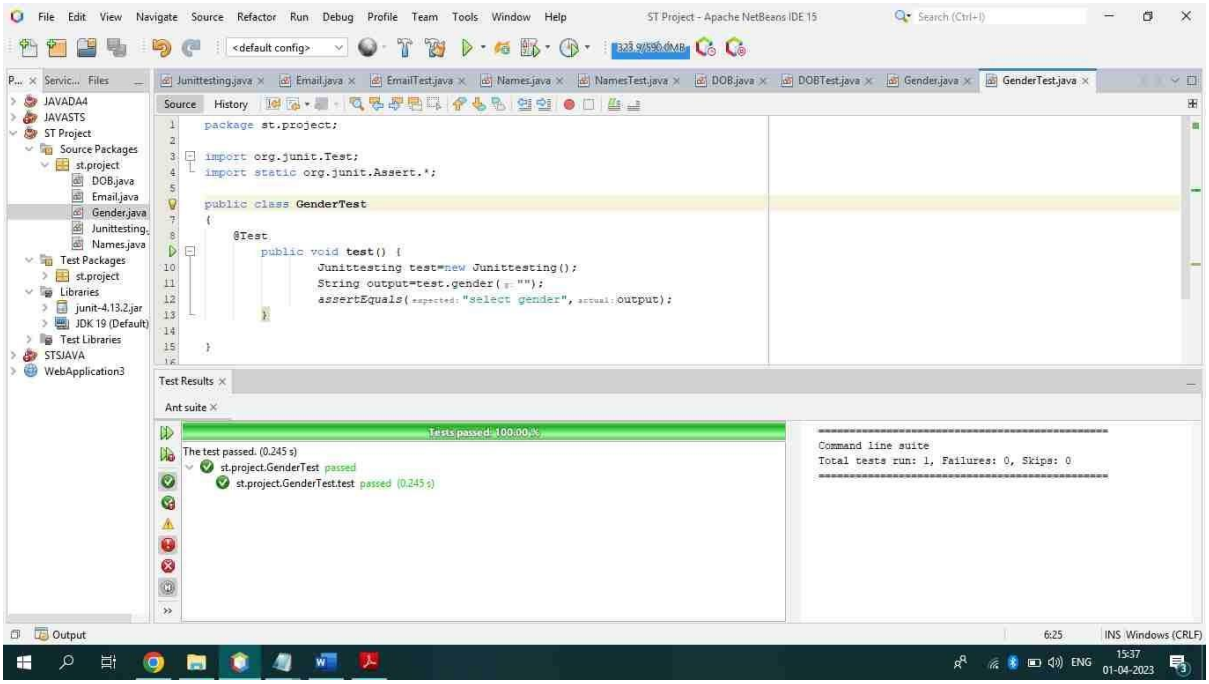


Test case id: T10
Input: 23- -
Expected output: select month, year

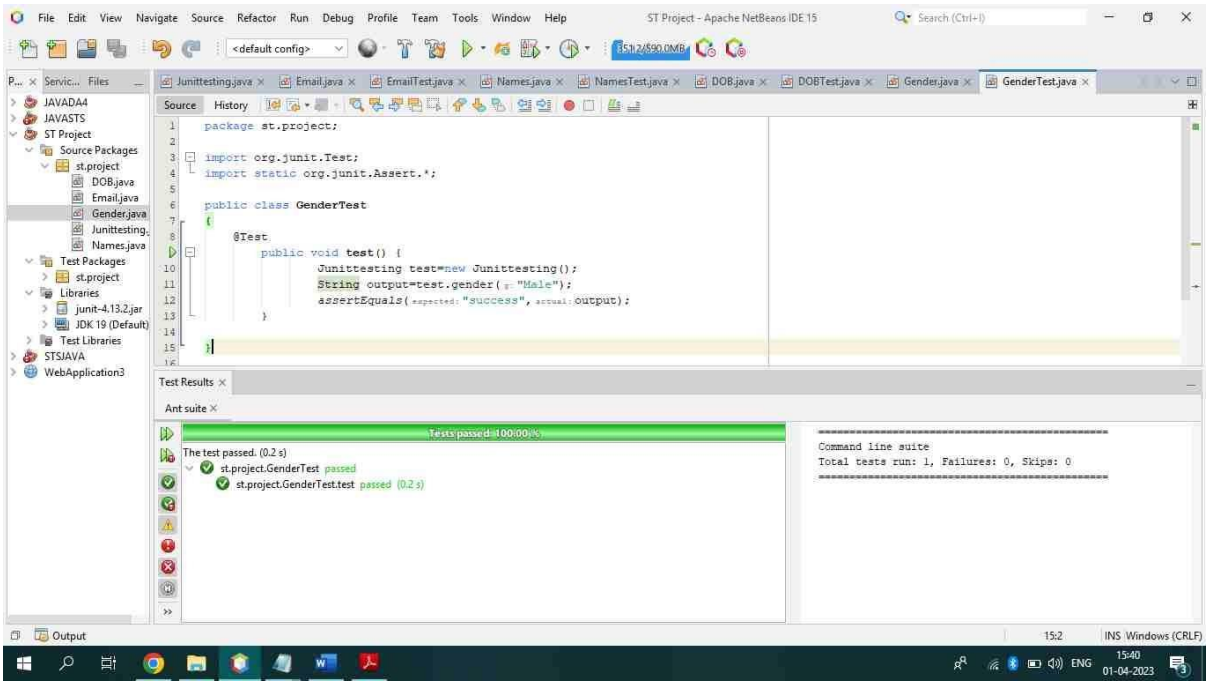


Unit 3: Gender

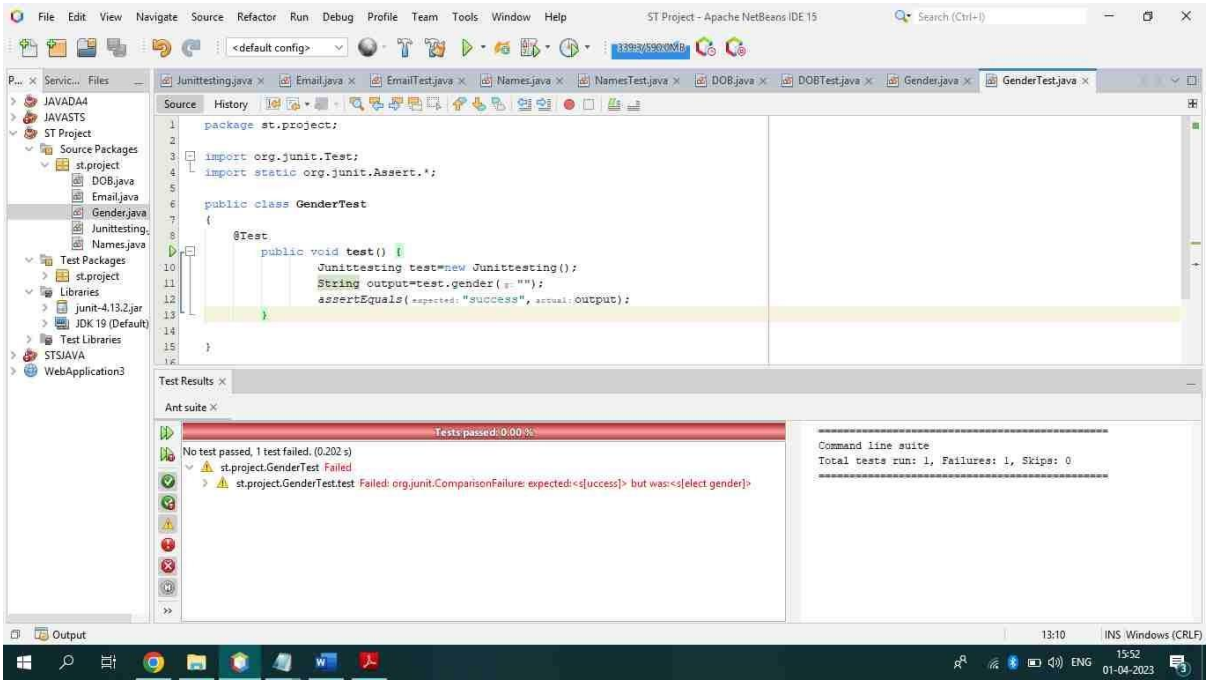
Test case id: T11
Input:
Expected output: select gender



Test case id: T12
Input: Male
Expected output: success

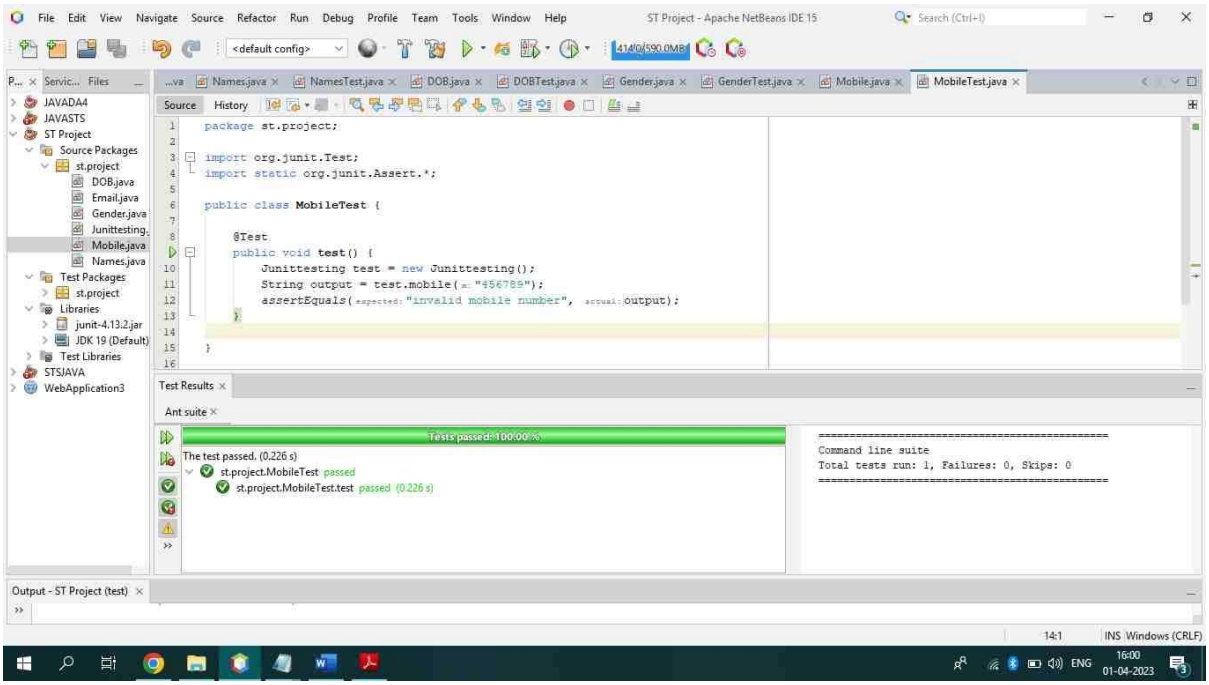


Test case id: T13(Failed Test Case)
Input:
Expected output: mobile number is required

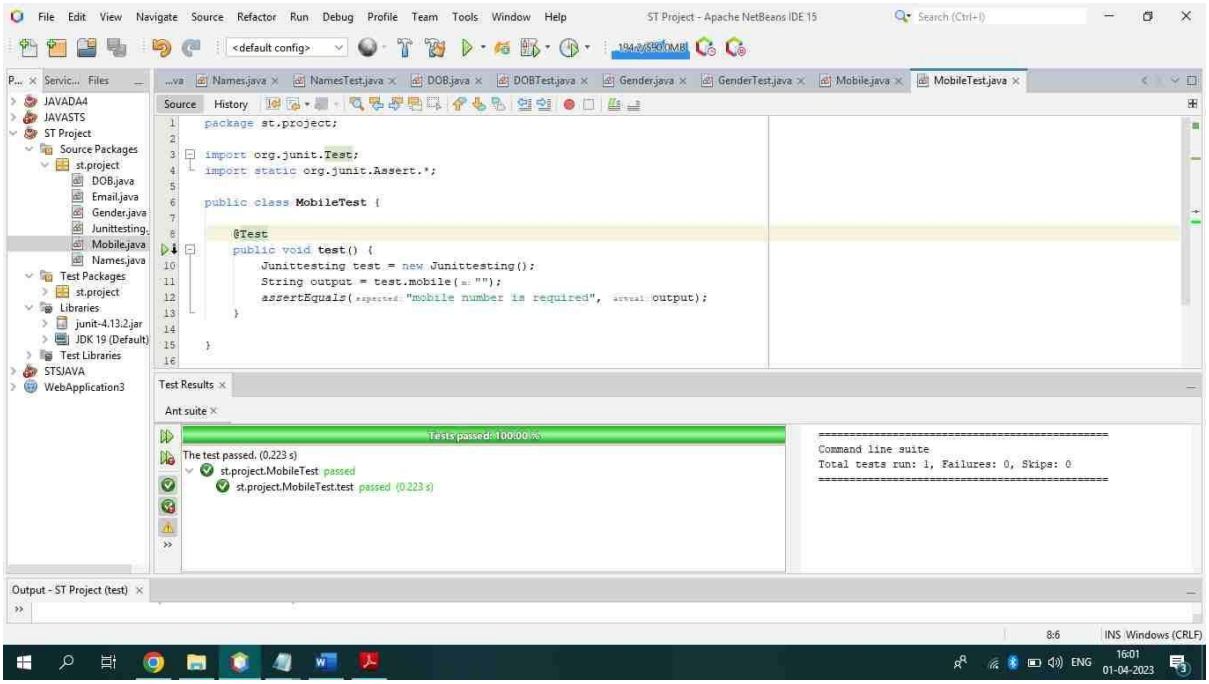


Unit 4: Mobile Number

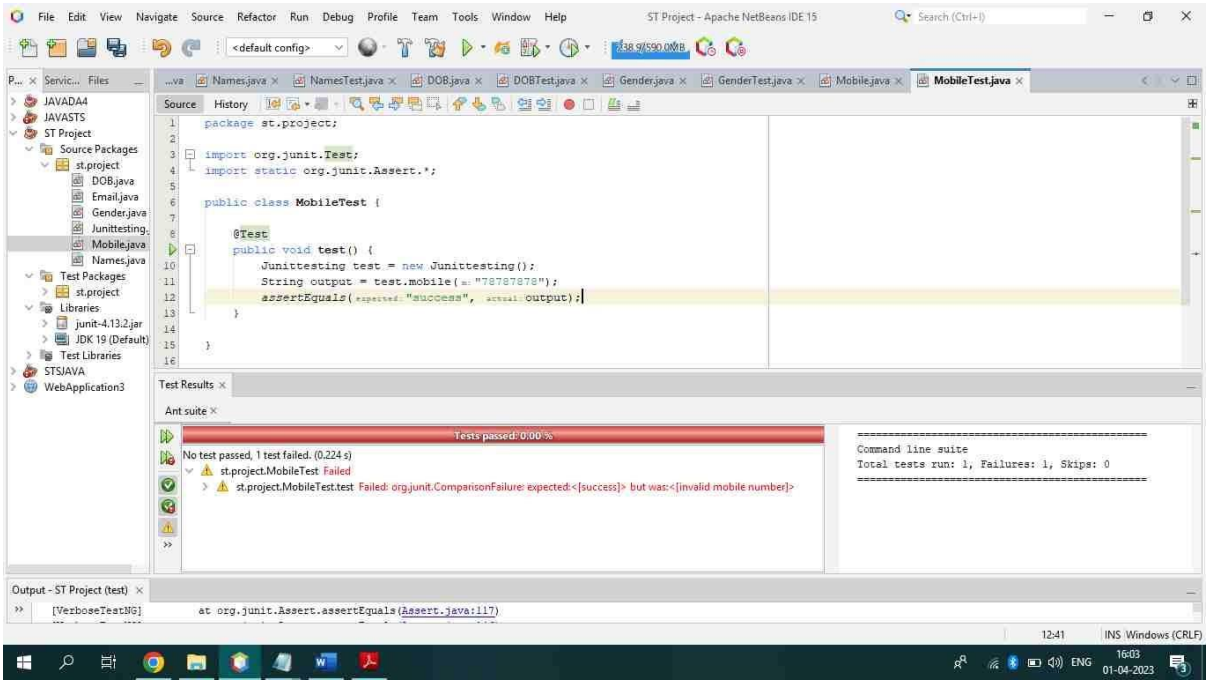
Test case id: T14
Input: 456789
Expected output: invalid mobile number



Test case id: T15
Input:
Expected output: mobile number is required

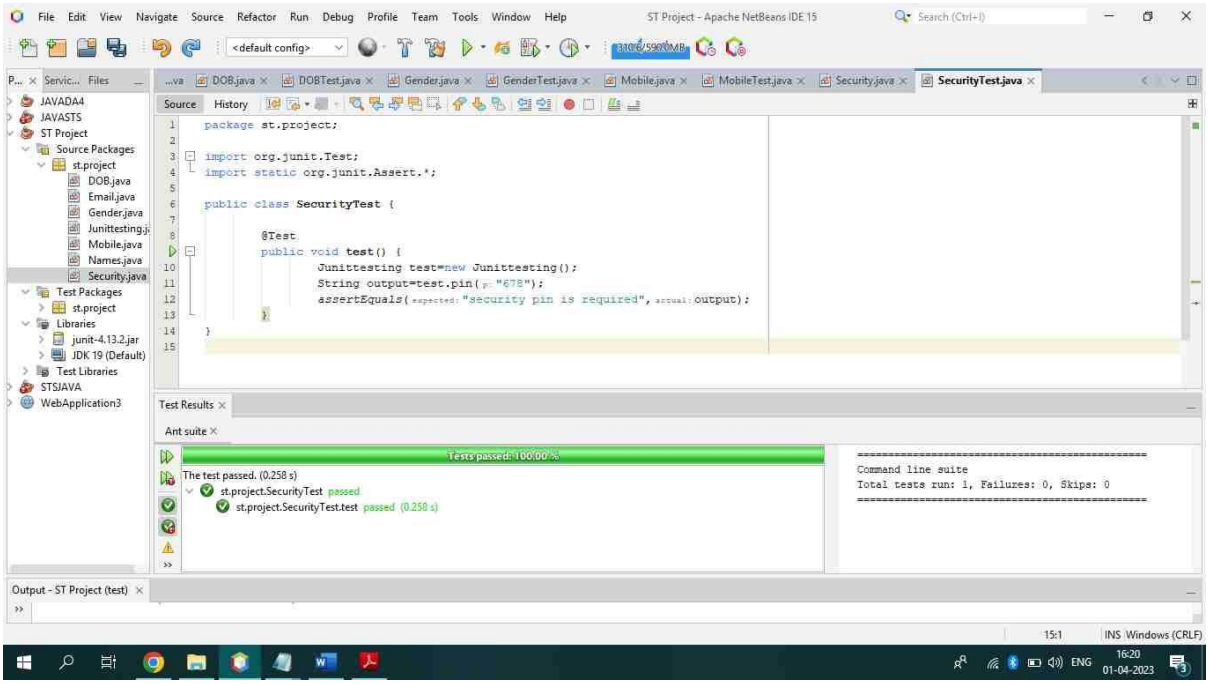


Test case id: T16 (Failed Test Case)
Input: 78787878
Expected output: Inavalid mobile number

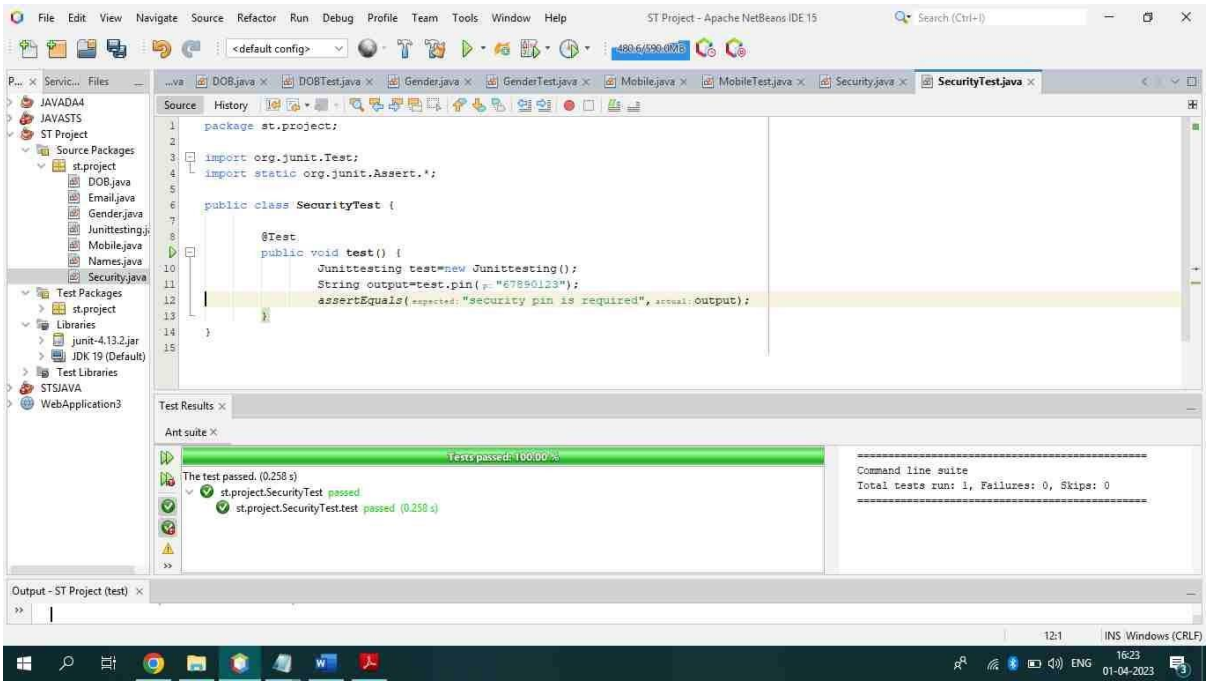


Unit 5: Security pin

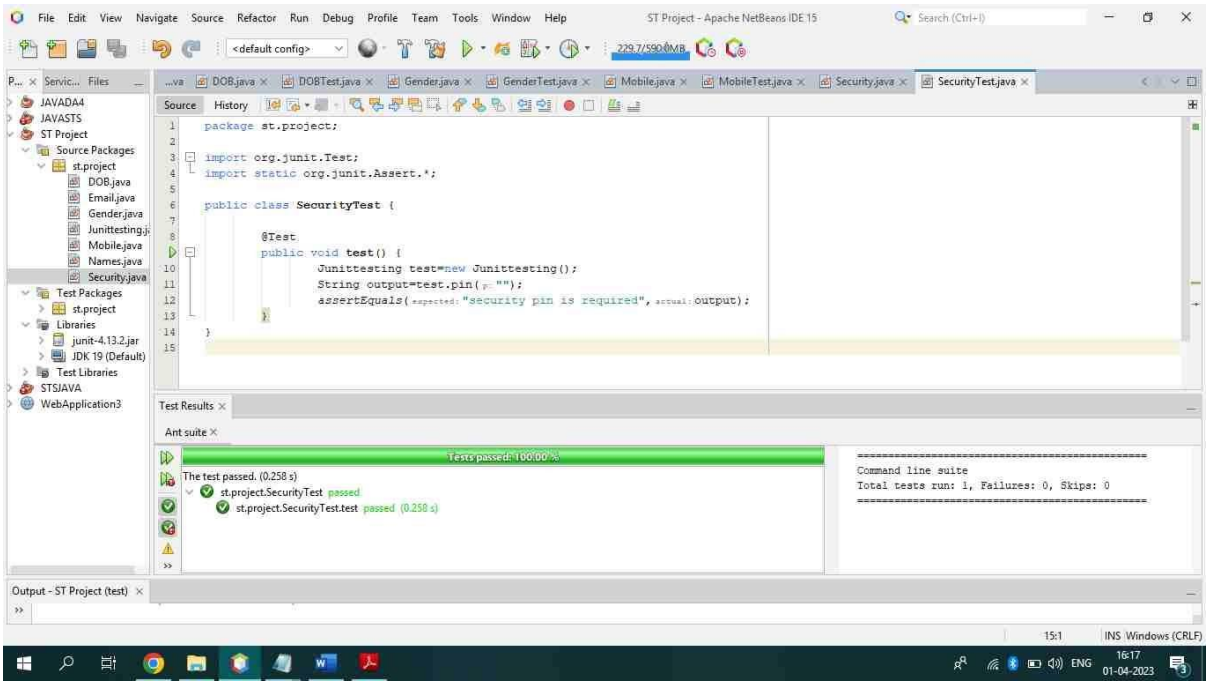
Test case id: T17
Input: 678
Expected output: pin must be 6 digits



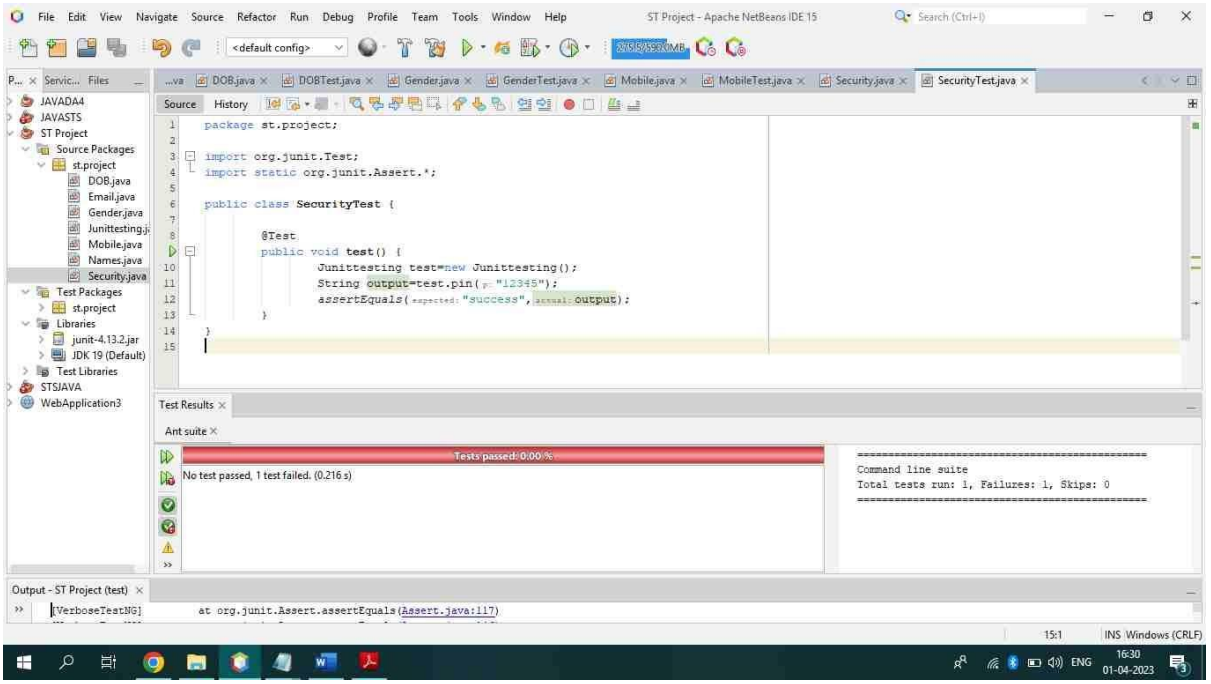
Test case id: T18
Input: 67890123
Expected output: pin must be 6 digits



Test case id: T19
Input:
Expected output: security pin is required

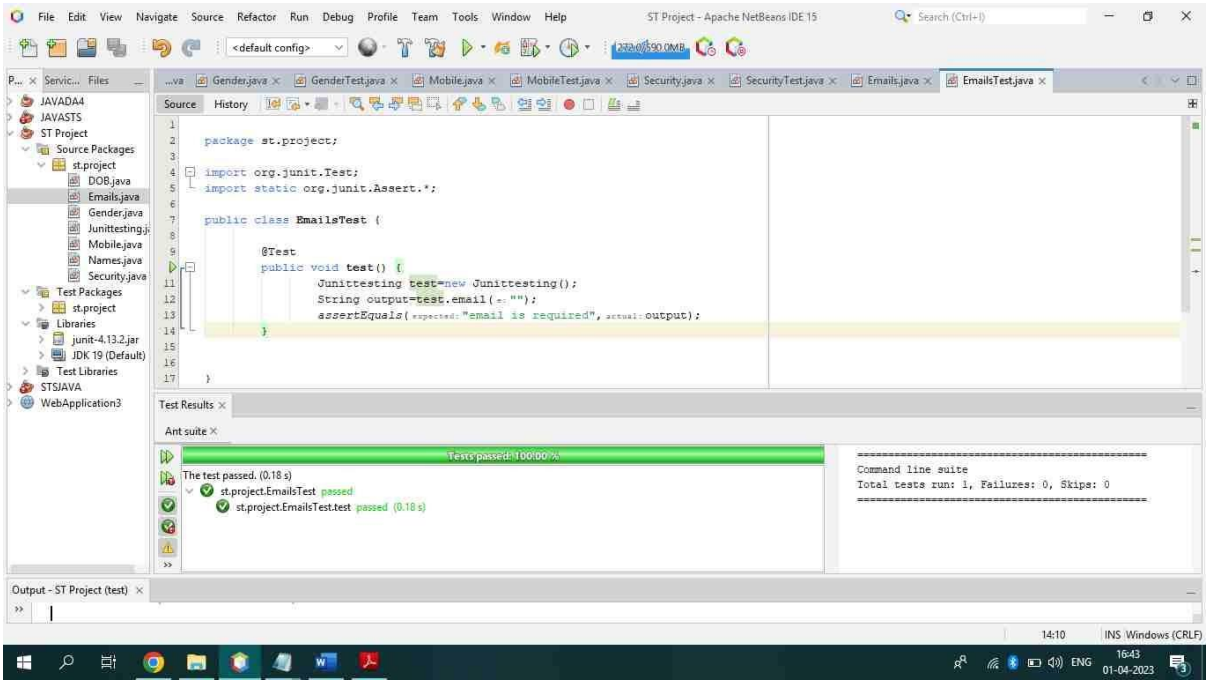


Test case id: T20 (Failed Test case)
Input: 12345
Expected output: success

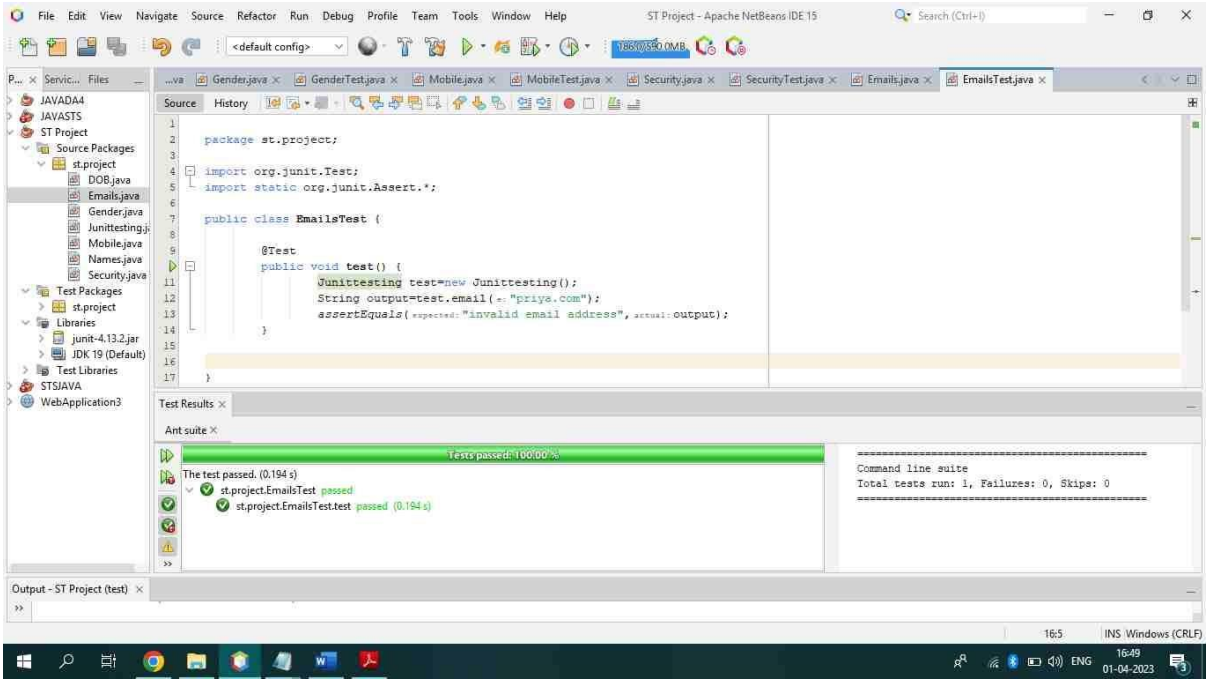


Unit 6: Email

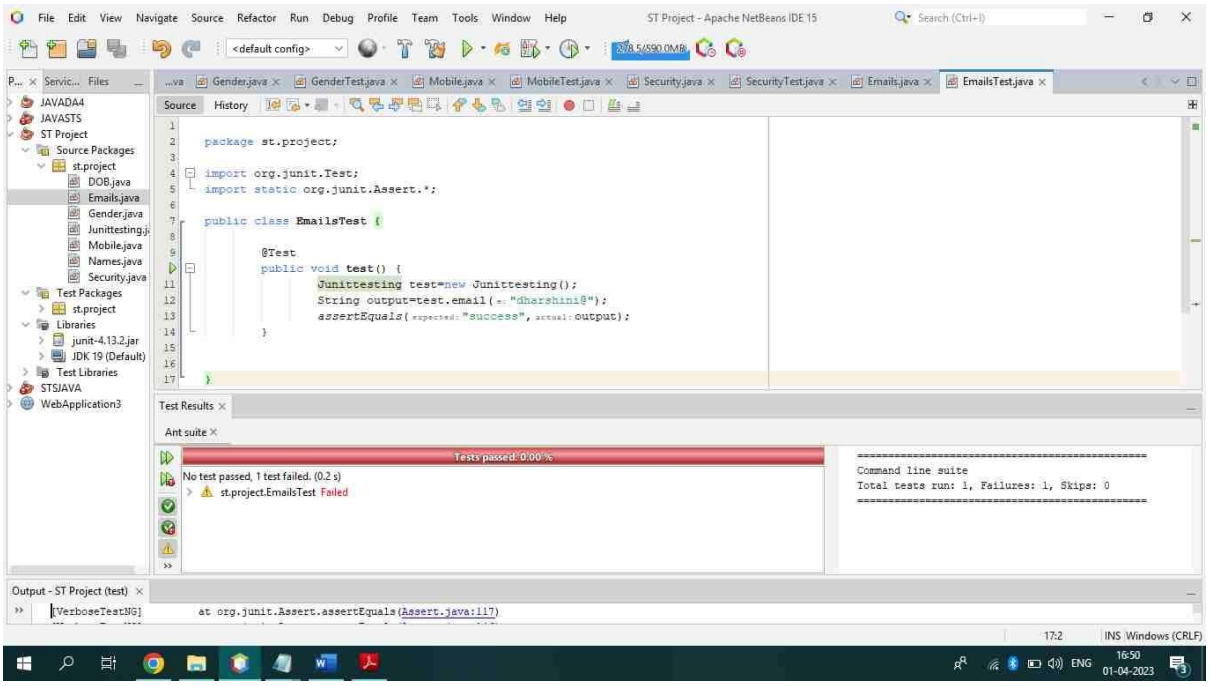
Test case id: T21
Input:
Expected output: email is required



Test case id: T22
Input: priya.com
Expected output: invalid email

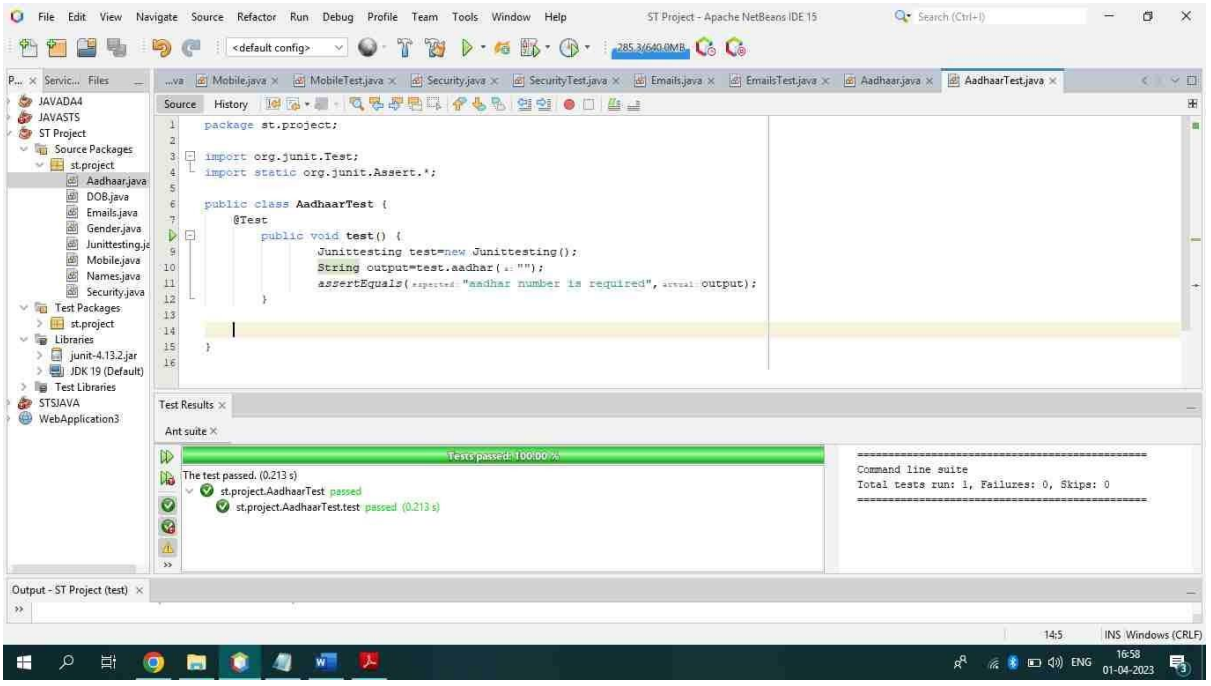


Test case id: T23 (Failed Test case)
Input: dharshini@
Expected output: invalid email

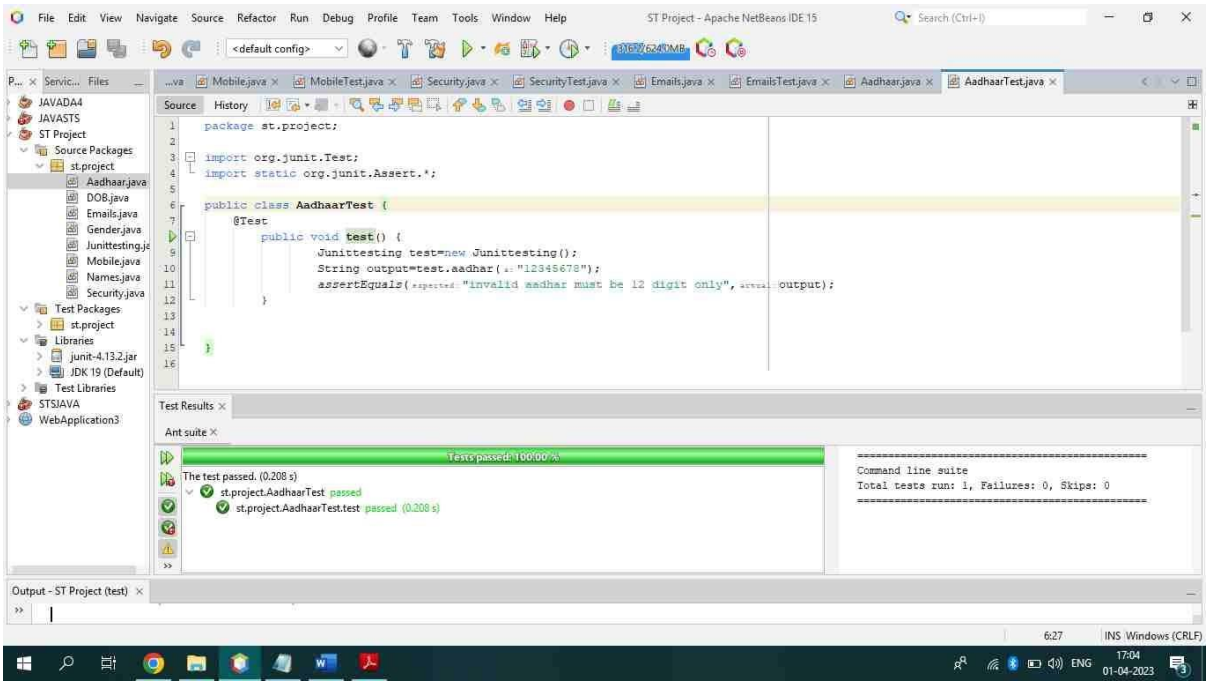


Unit 7: Aadhaar Number

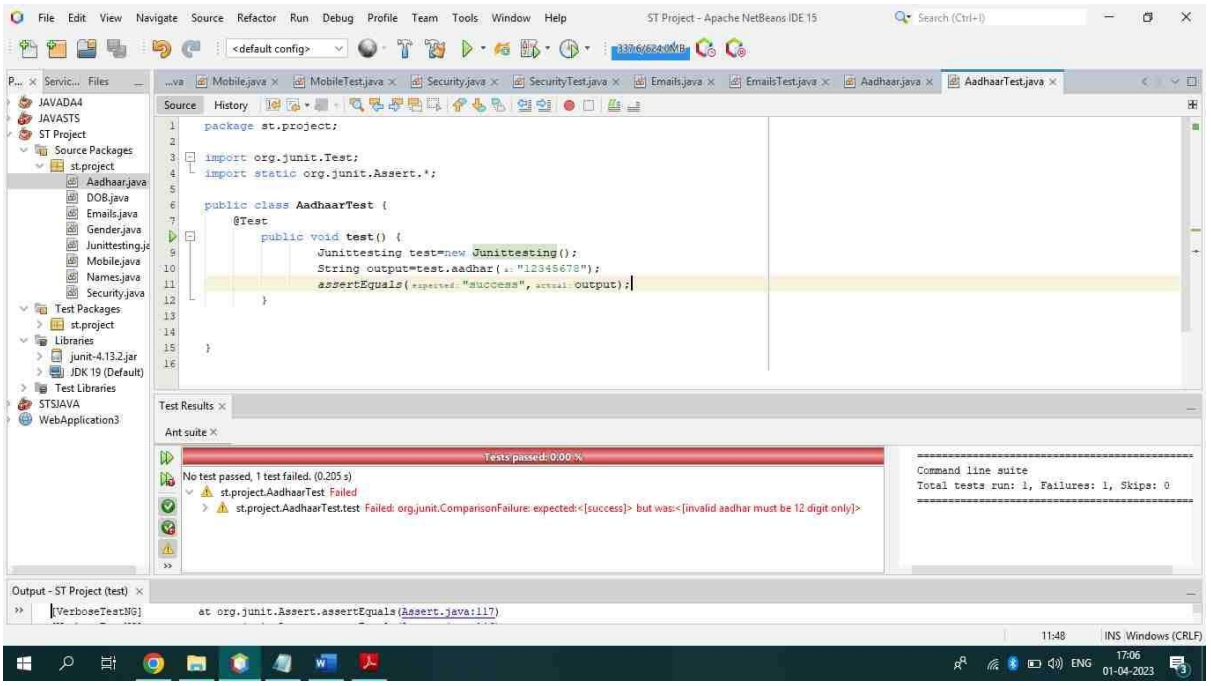
Test case id: T24
Input:
Expected output: aadhar number is required



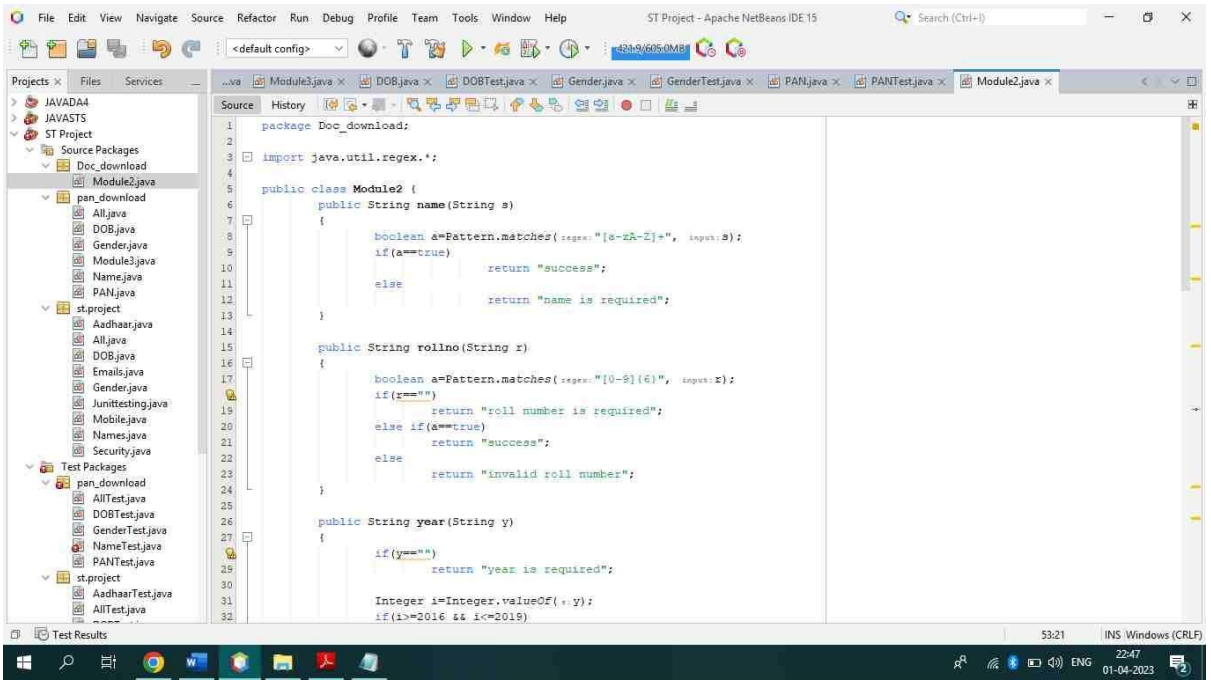
Test case id: T25
Input: 12345678
Expected output: invalid aadhar must be 12 digit only



Test case id: T26(Failed Test case)
Input: 12345678
Expected output: invalid aadhar number

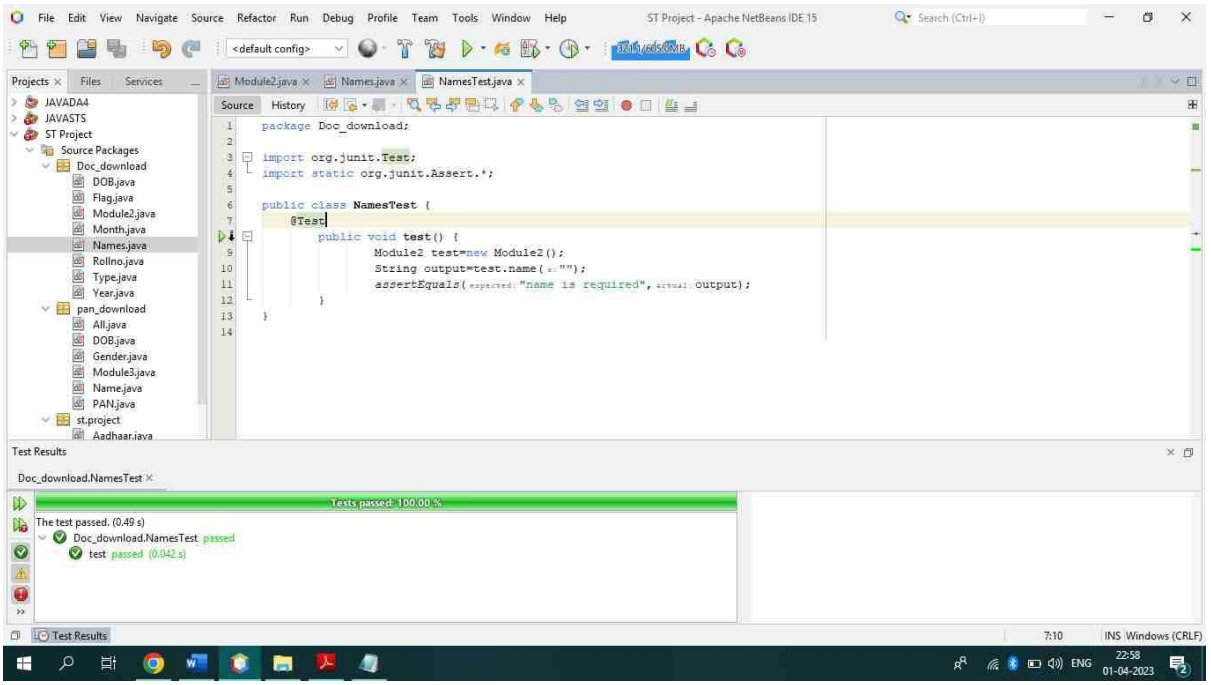


MODULE 2:Marksheet Download



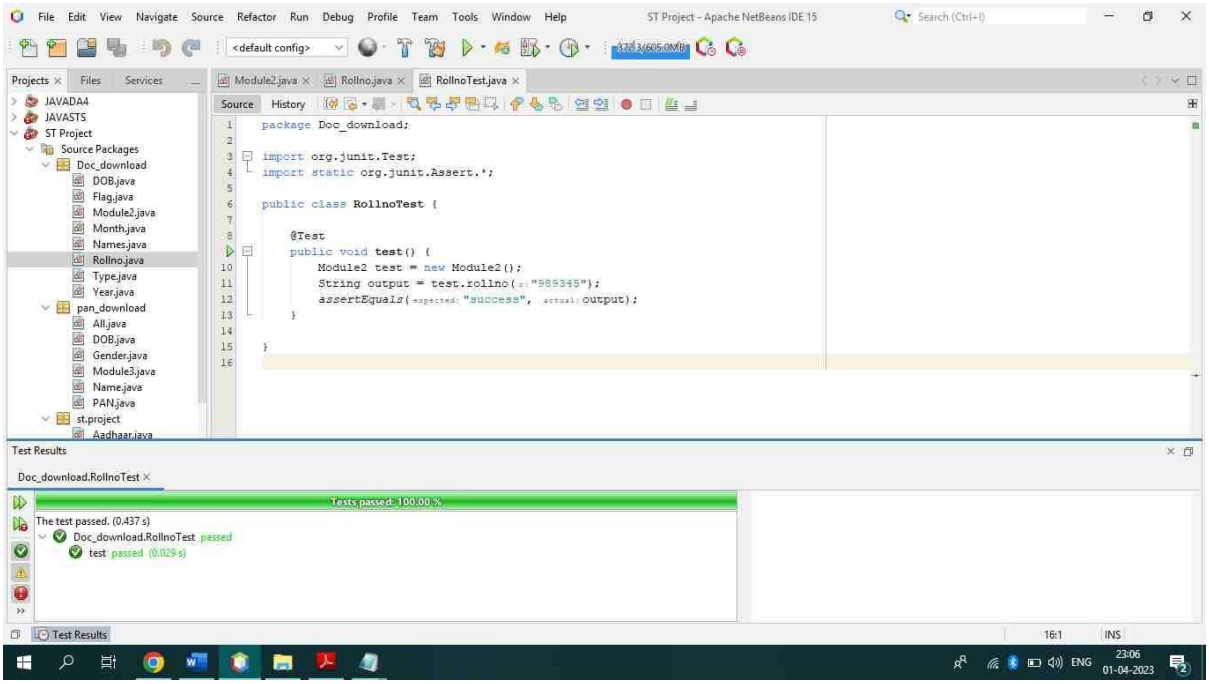
Unit 1: Name

Test case id: TC1
Input:
Expected output: name is required

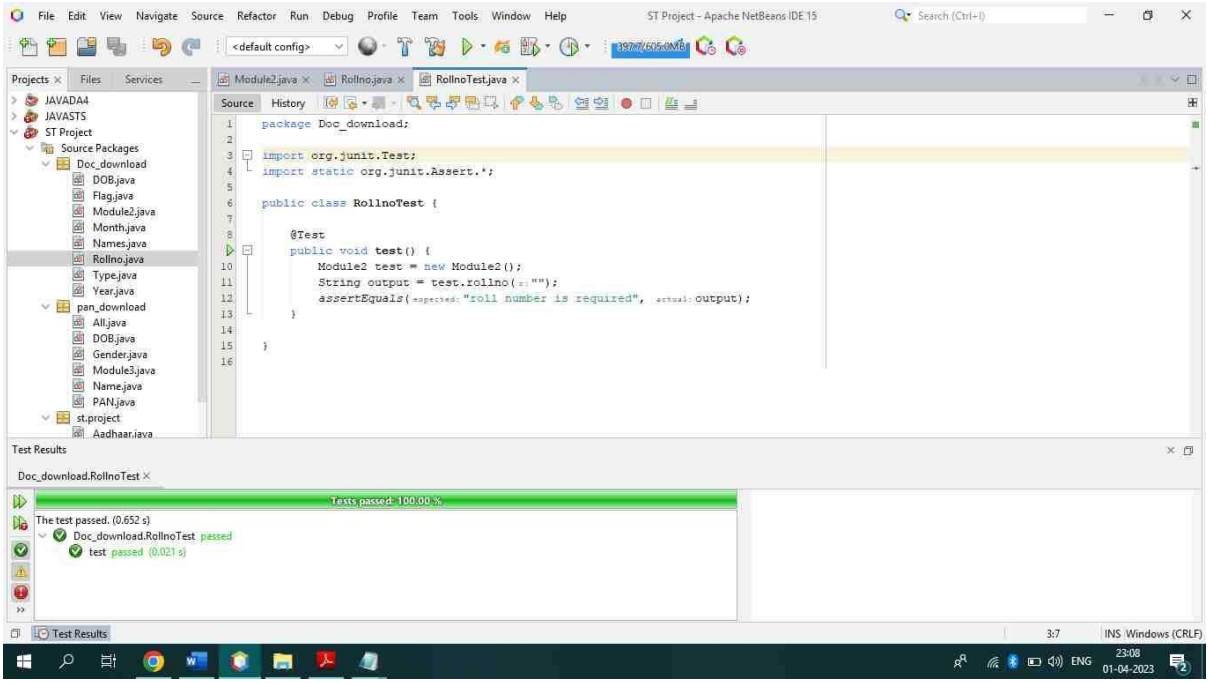


Unit 2: Roll No

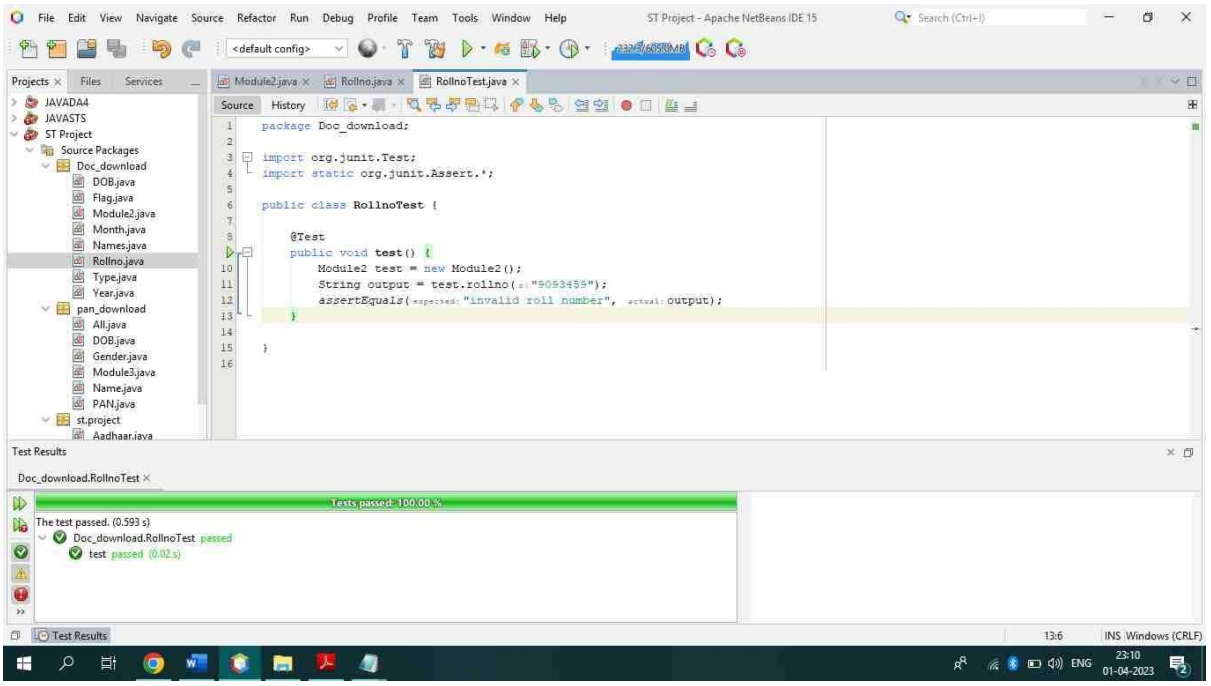
Test case id: TC2
Input: 909345
Expected output: success



Test case id: TC3
Input:
Expected output: roll number is required

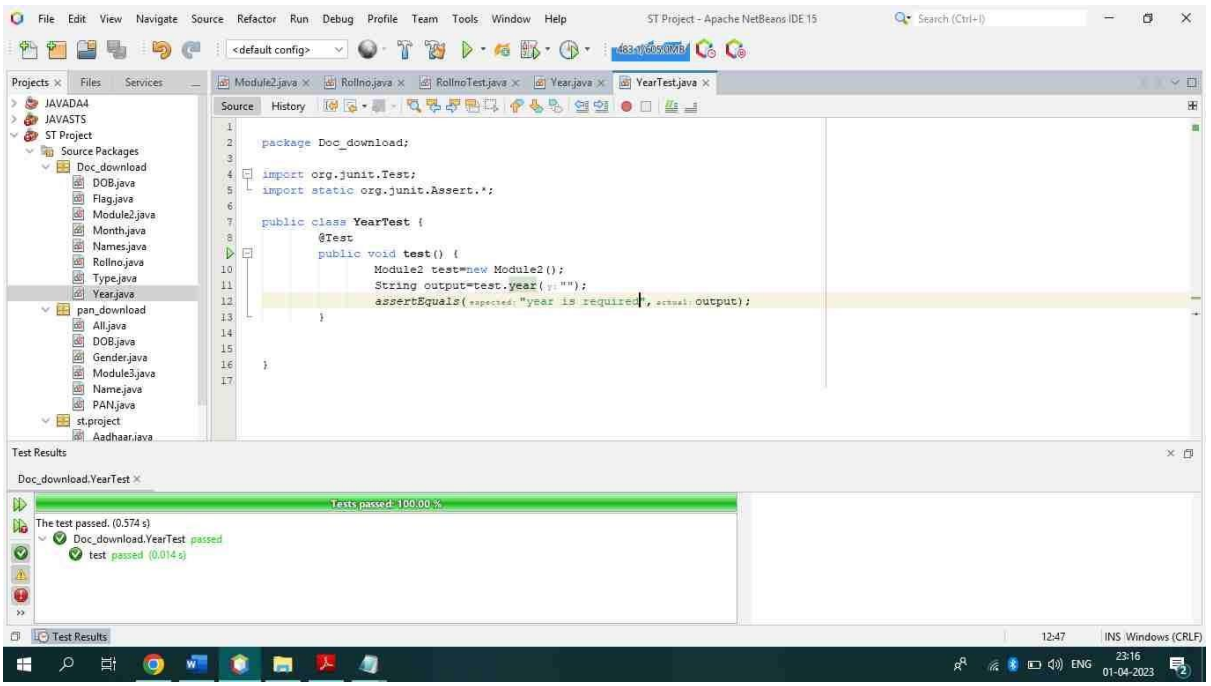


Test case id: TC4
Input: 9093459
Expected output: invalid roll number

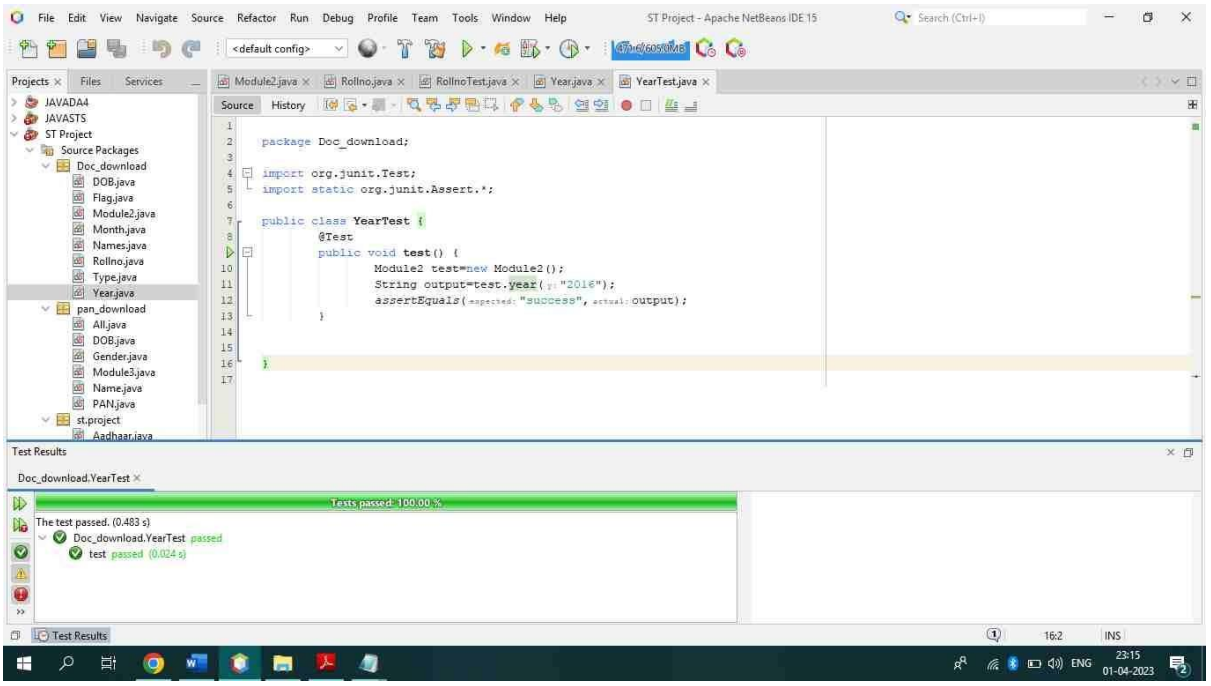


Unit 3: Year

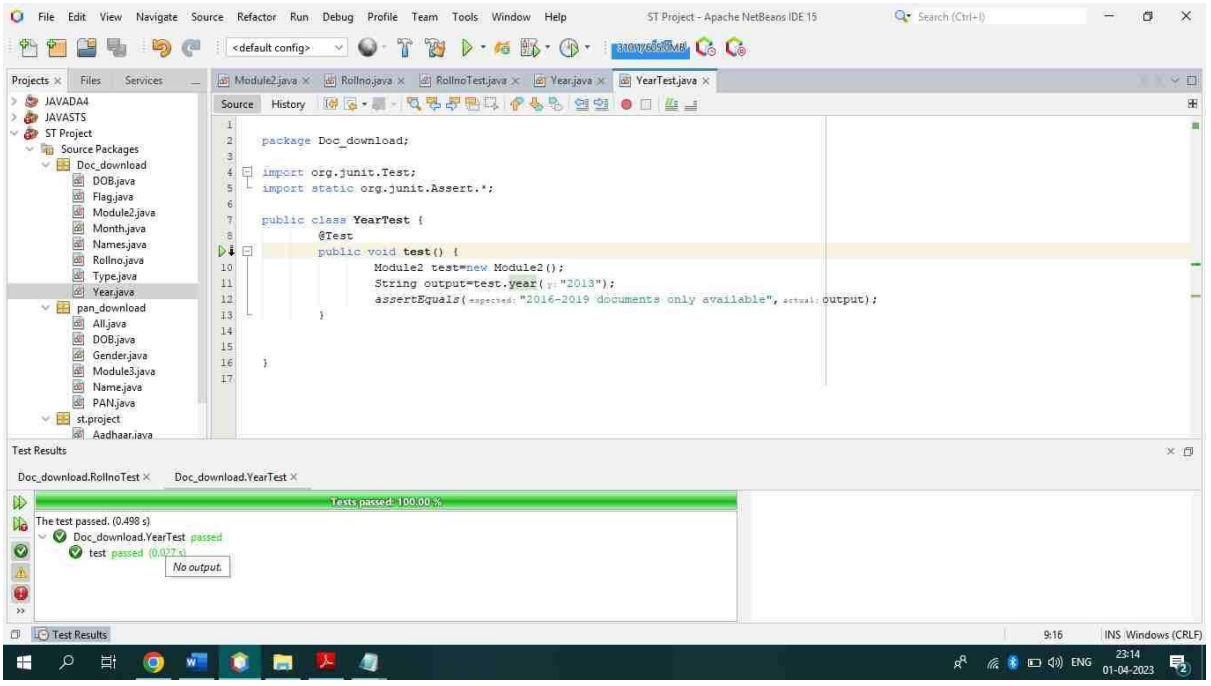
Test case id: TC5
Input:
Expected output: year is required



Test case id: TC6
Input: 2016
Expected output: success

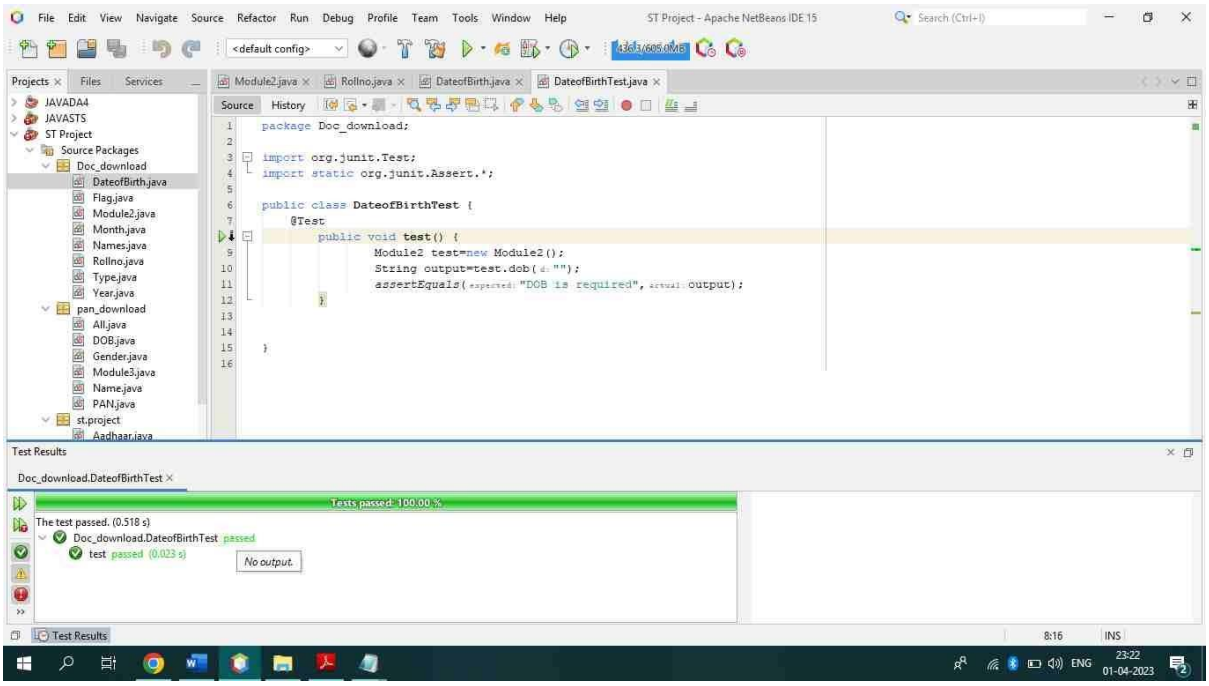


Test case id: TC7
Input: 2013
Expected output: 2016-2019 documents only available



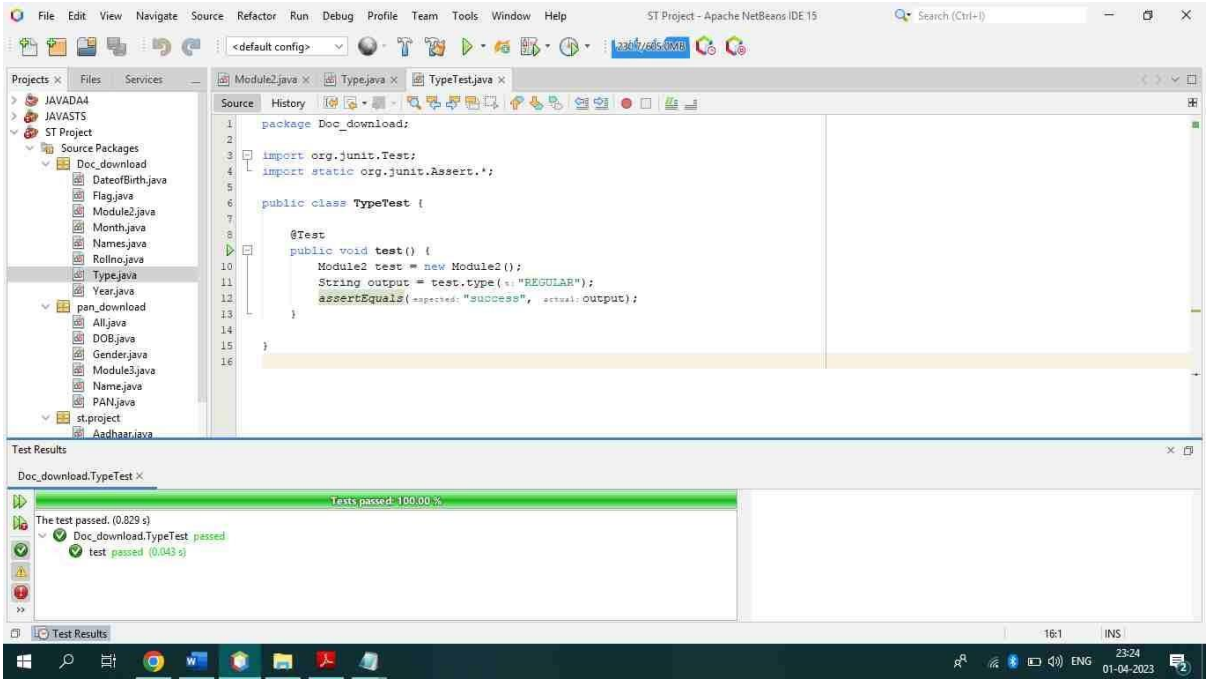
Unit 4: DOB

Test case id: TC8
Input:
Expected output: DOB is required

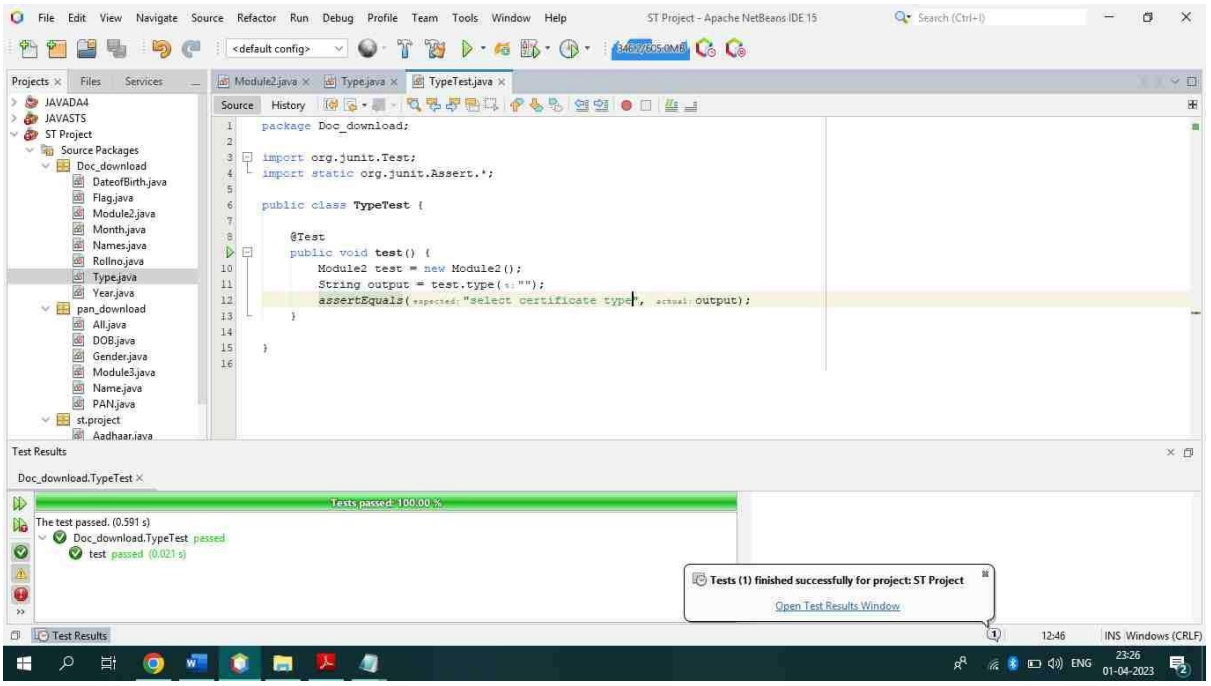


Unit 5: Certificate Type

Test case id: TC8
Input: regular
Expected output: success

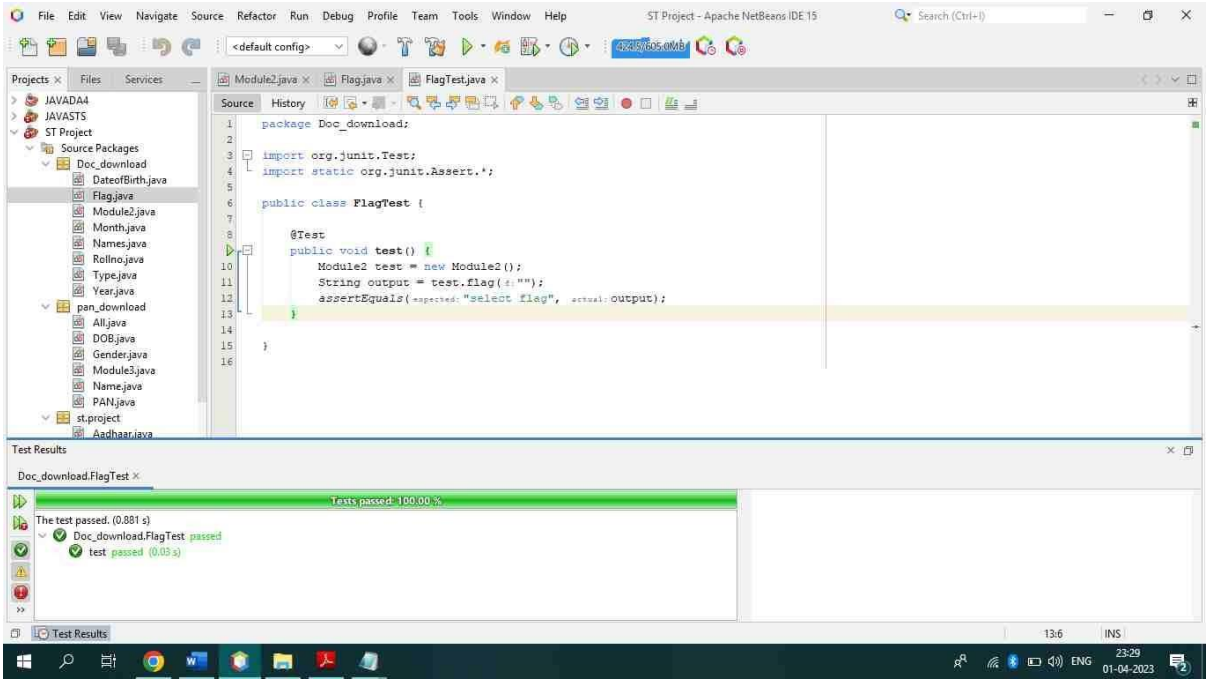


Test case id: TC9
Input:
Expected output: select certificate type

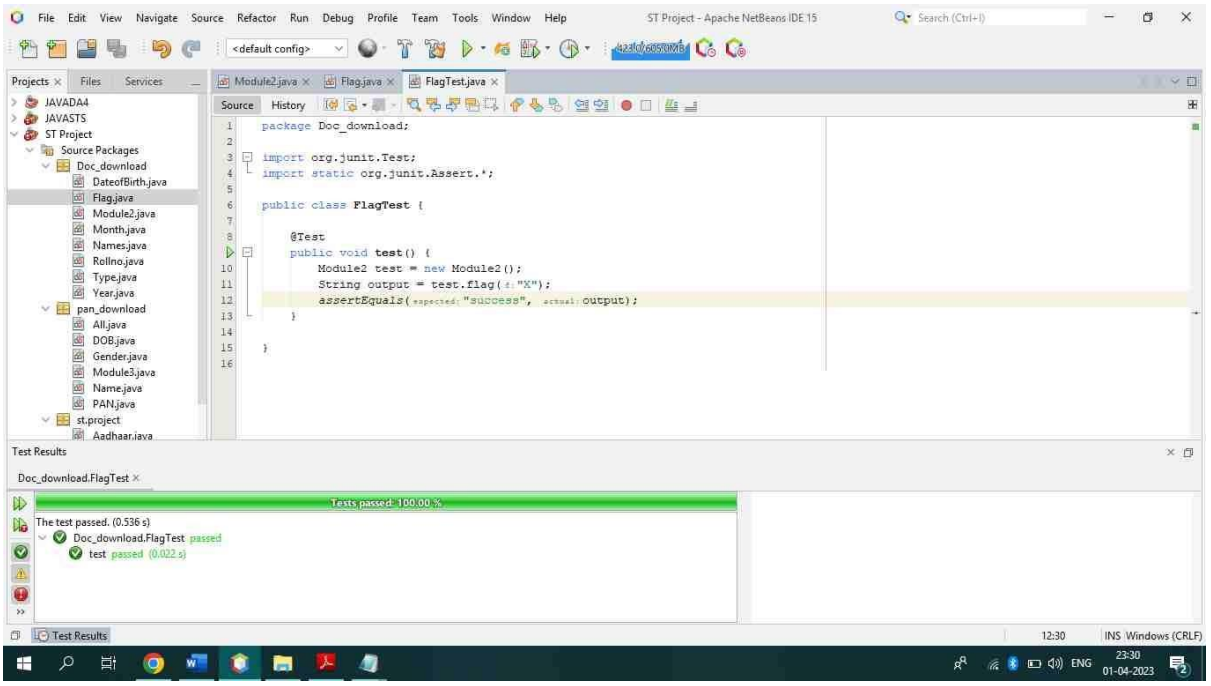


Unit 6: Flag

Test case id: TC10
Input:
Expected output: select flag

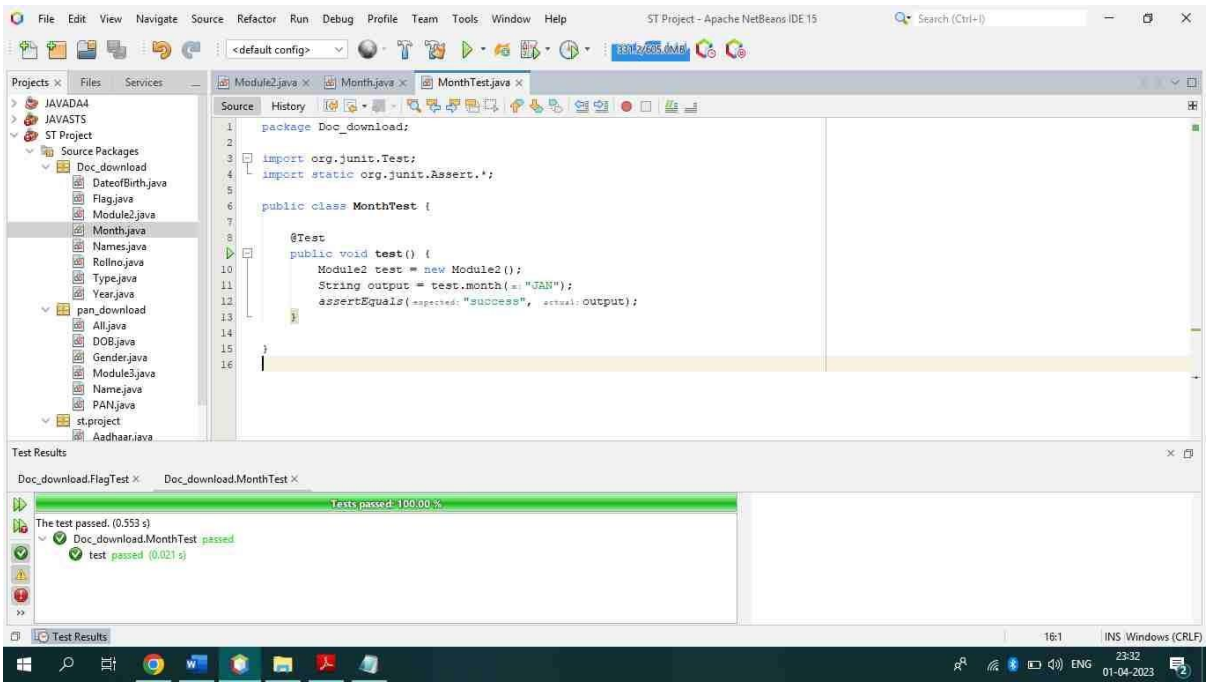


Test case id: TC11
Input:
Expected output: success

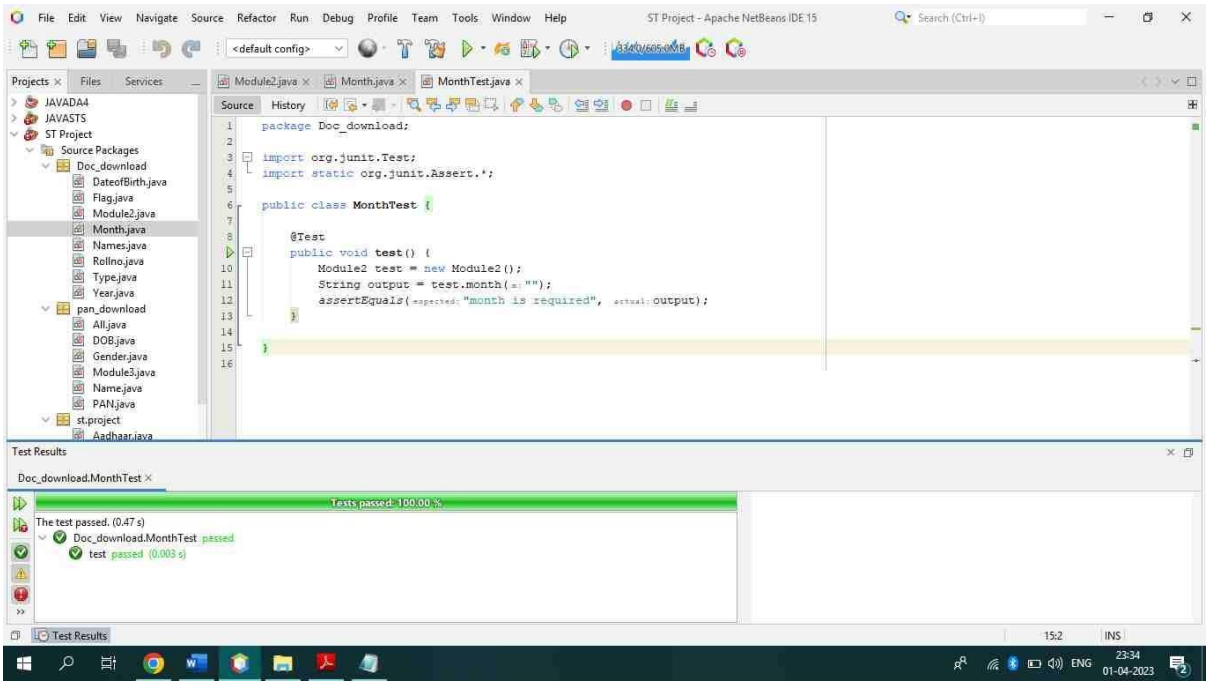


Unit 7: Month

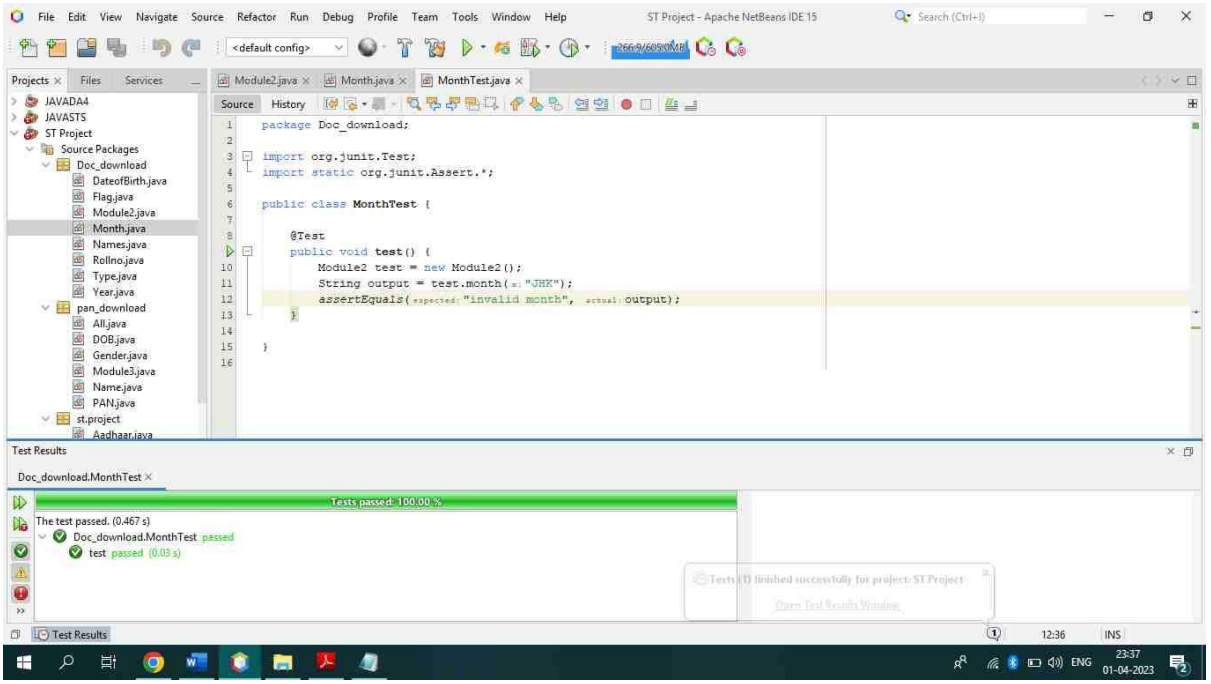
Test case id: TC12
Input: JAN
Expected output: success



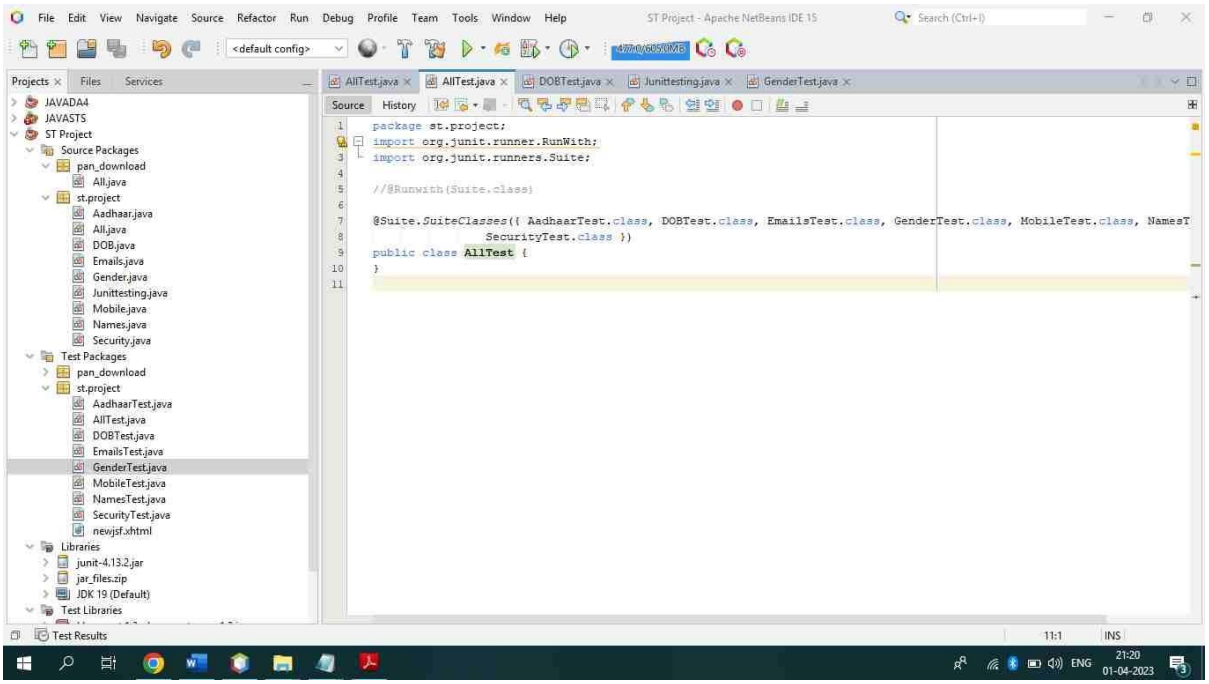
Test case id: TC13
Input:
Expected output: month is required

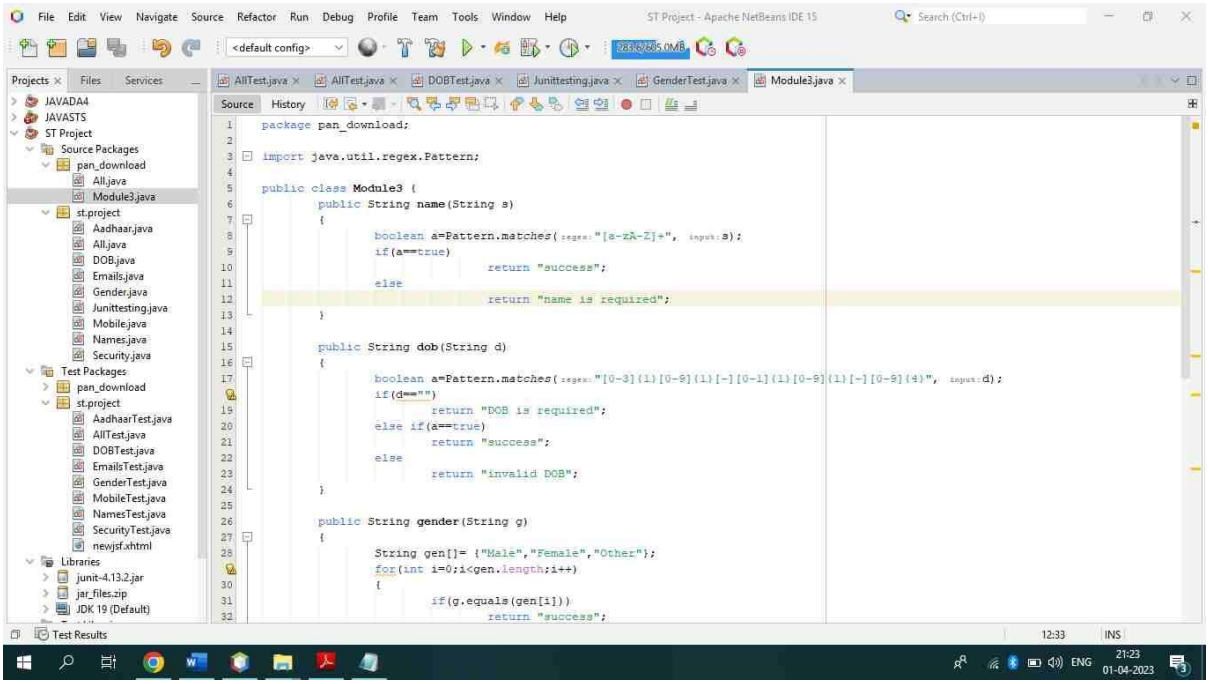


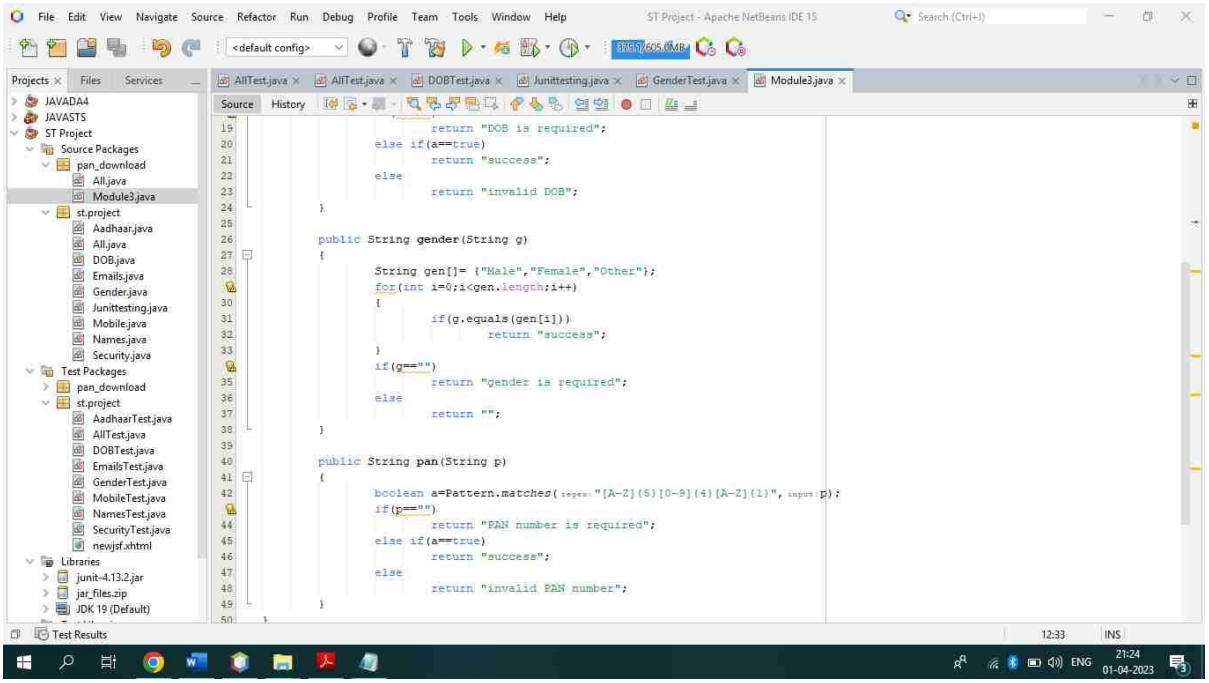
Test case id: TC14
Input: JHK
Expected output: invalid month



MODULE 3:PAN card DOWNLOAD

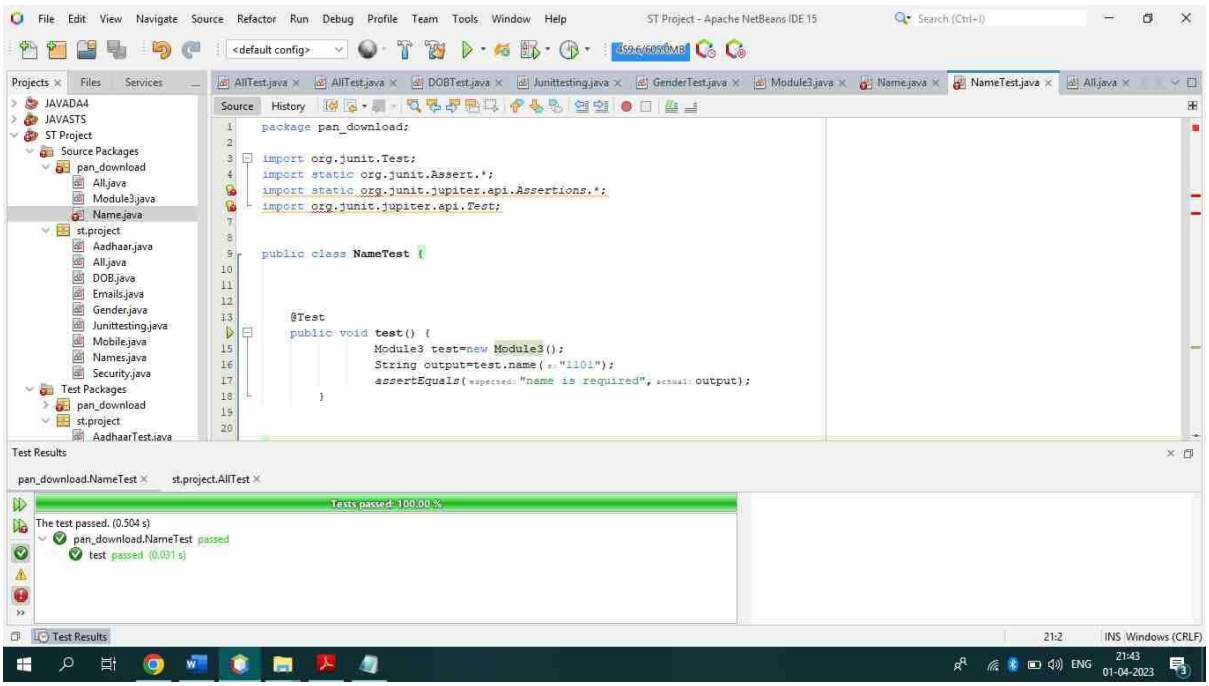




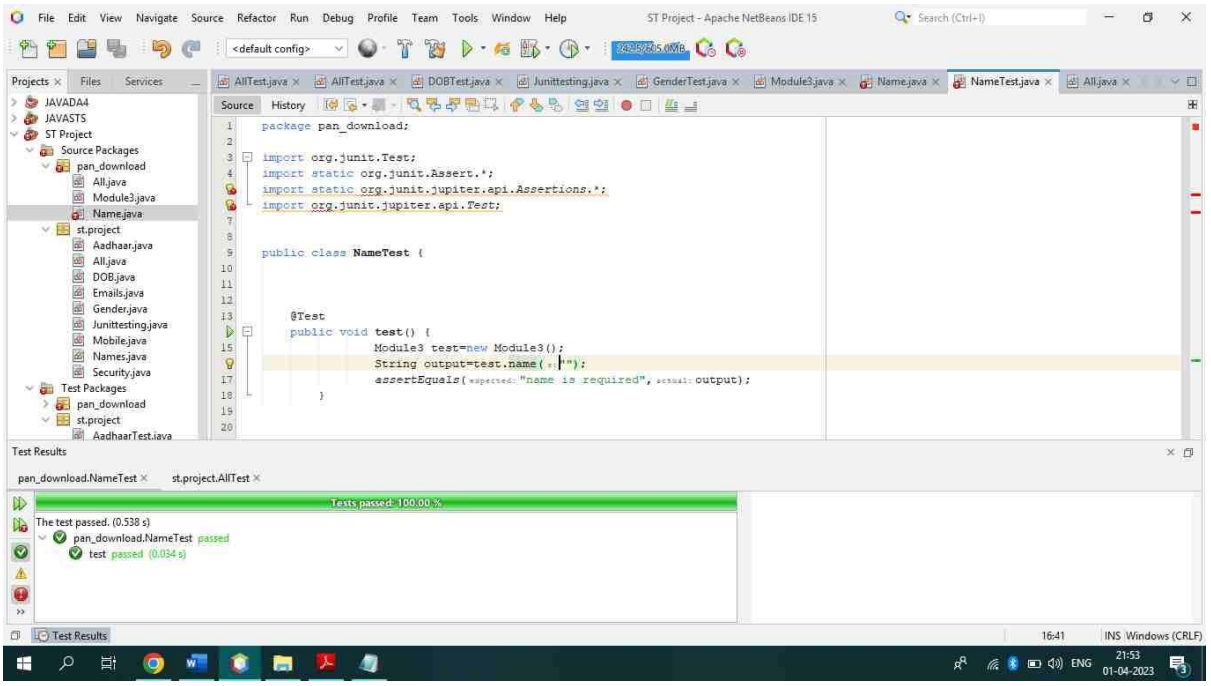


Unit 1: Name

Test case id: T01
Input: 123
Expected output: “name is required”

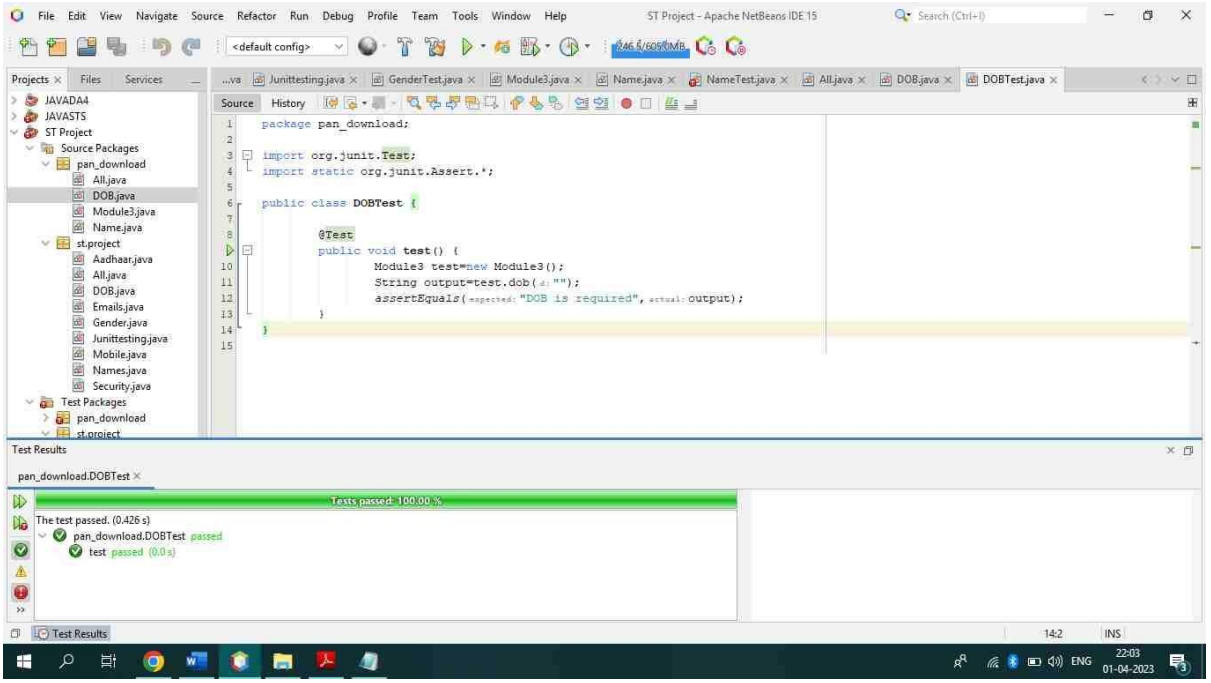


Test case id: T02
Input:
Expected output: name is required.



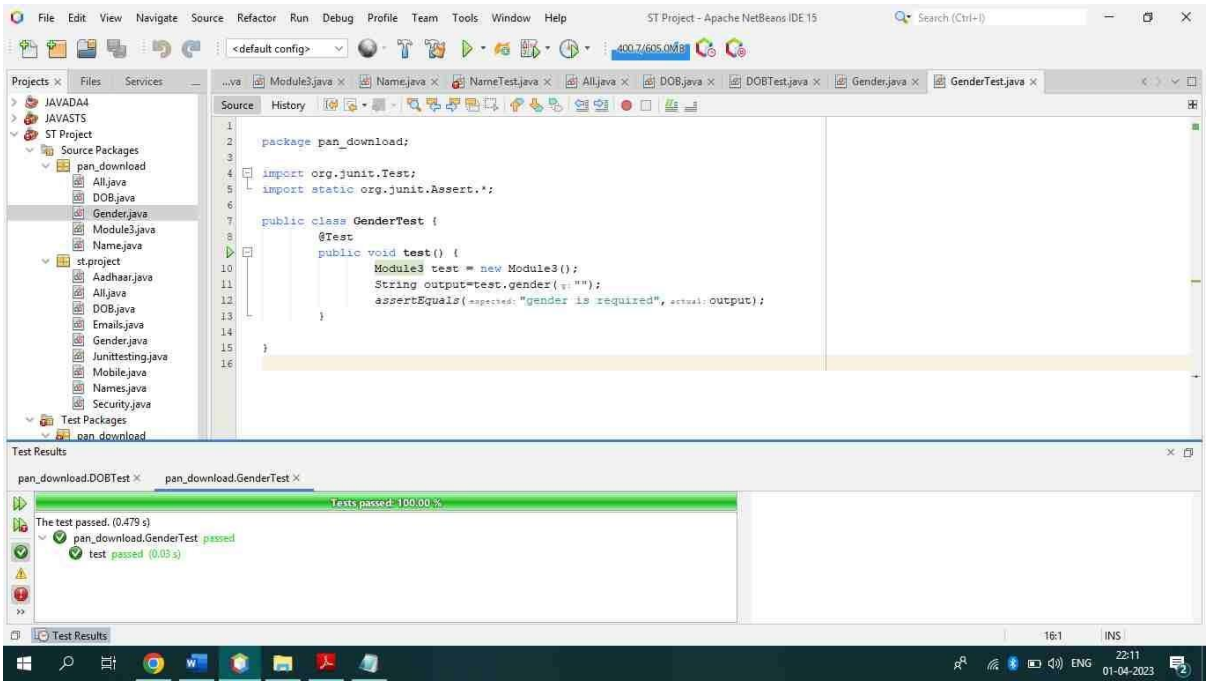
Unit 2: DOB

Test case id: T03
Input:
Expected output: DOB is required



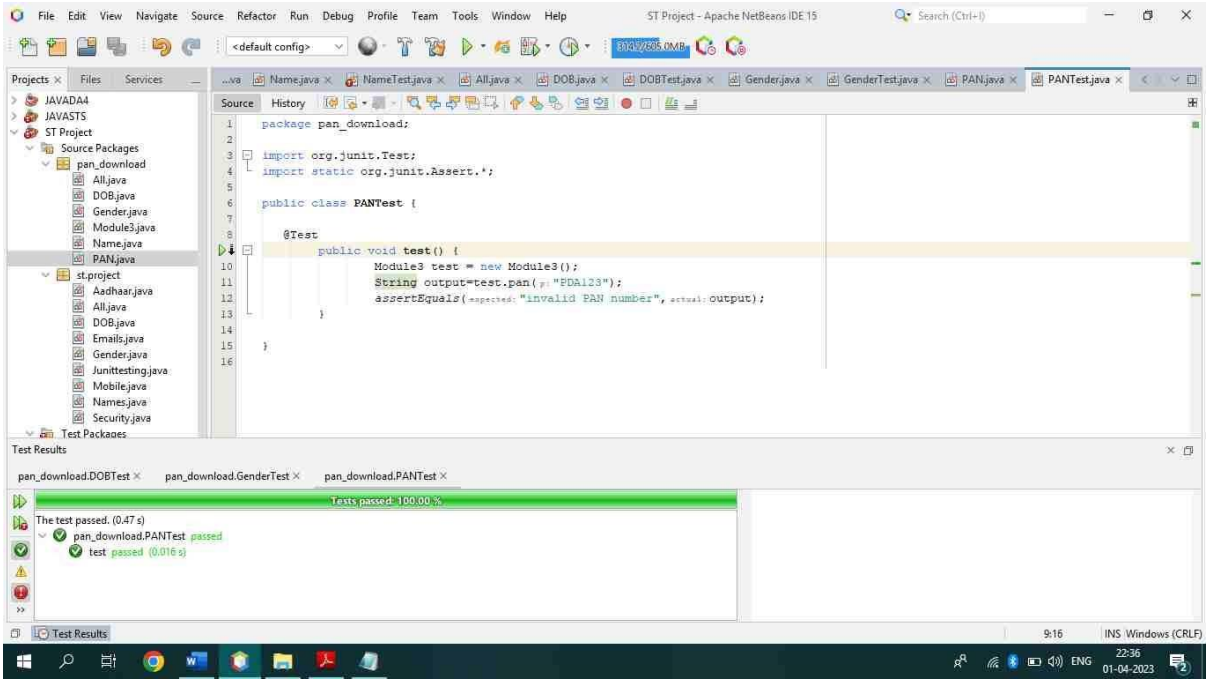
Unit 3: Gender

Test case id: T03
Input:
Expected output: gender is required.

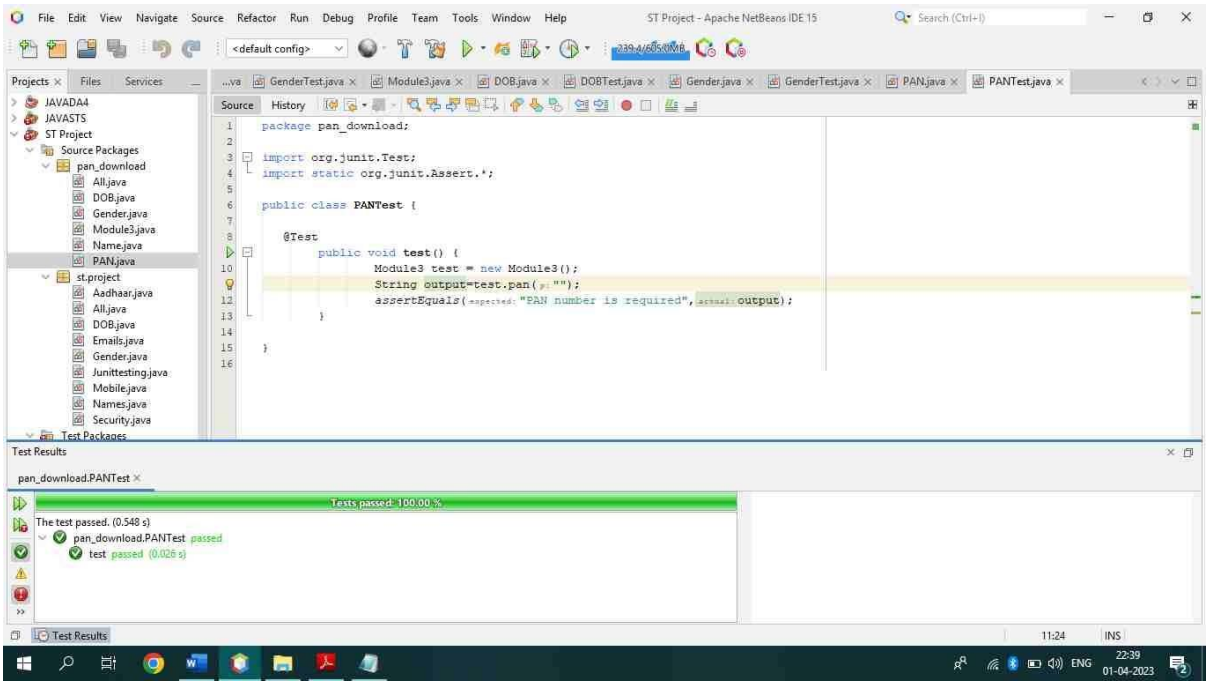


Unit 4: PAN number

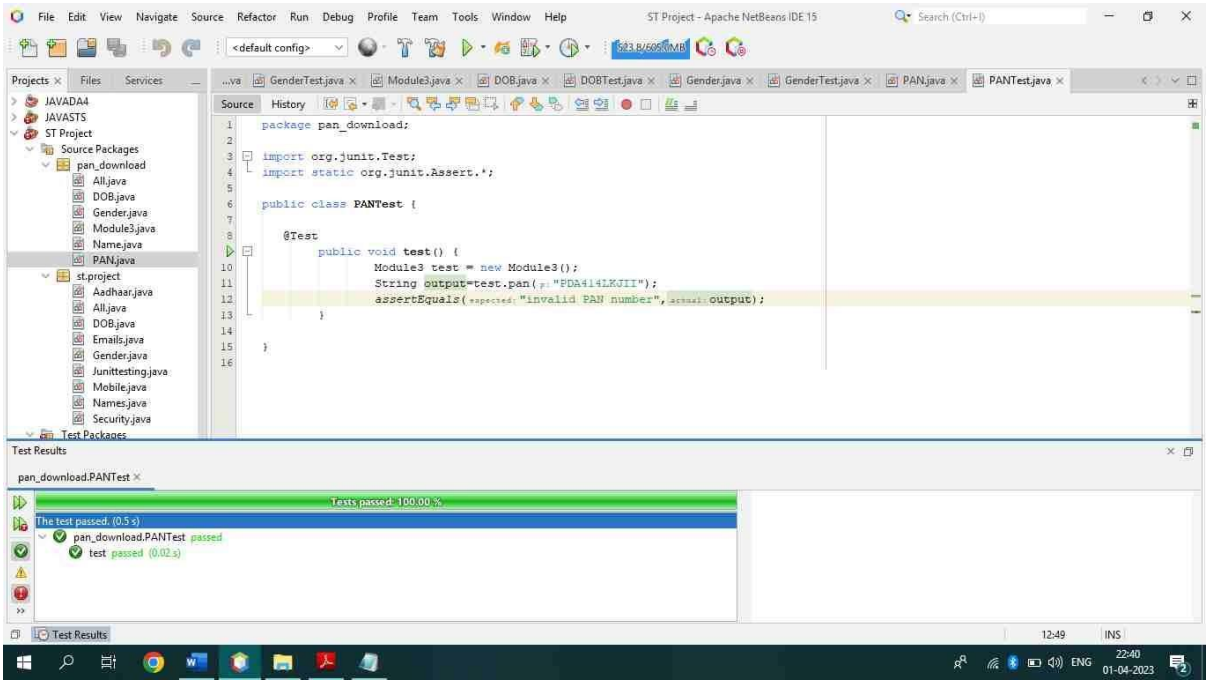
Test case id: T04
Input: PDA414
Expected output: invalid PAN number



Test case id: T05
Input:
Expected output: PAN number is required



Test case id: T06
Input: PDA414LKJII
Expected output: invalid PAN number



Conclusion :

We have created Test Case report and Here we successfully tested Digilocker website with the help of Junit testing tool and netbeans. We successfully added some bugs to to code and did testing with junit tool.