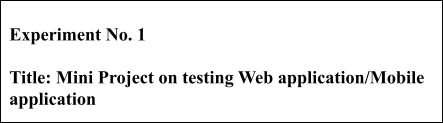
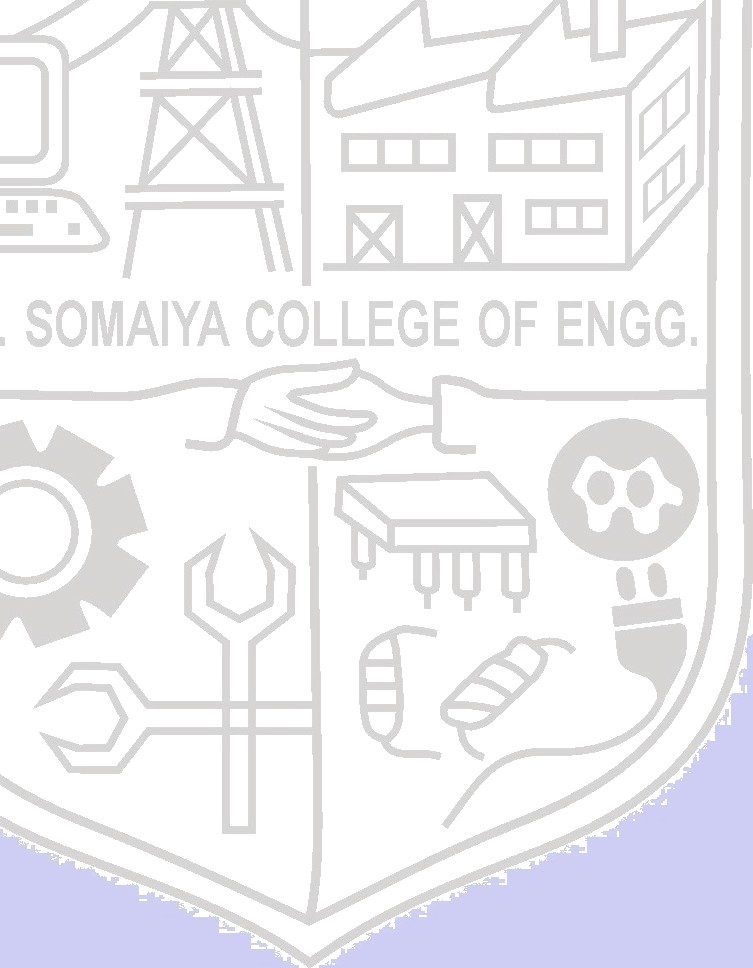


KJSCE/IT/LYBTech/SEMVII/STQA/2024-25





# Batch:B1 Roll No.: 16010421119 Experiment No.:1

**Aim:** To develop a Web application/Mobile application and test it by using open source software testing tools.

**Resources needed:** Internet, Open source testing tool, Application development software

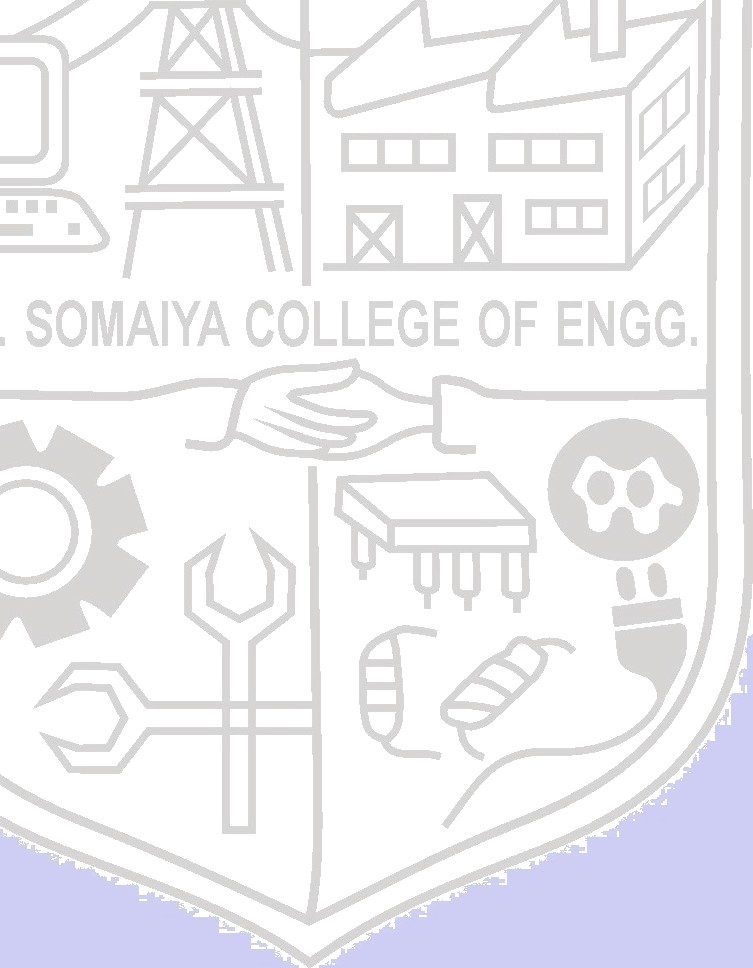
# Theory:

In [software testing](https://en.wikipedia.org/wiki/Software_testing), test automation is the use of special [software](https://en.wikipedia.org/wiki/Software) (separate from the software being tested) to control the execution of tests and the comparison of actual outcomes with predicted outcomes. Test automation can automate some repetitive but necessary tasks in a formalized testing process already in place, or add additional testing that would be difficult to perform manually.

Some [software testing](https://en.wikipedia.org/wiki/Software_testing) tasks, such as extensive low-level interface [regression testing](https://en.wikipedia.org/wiki/Regression_testing), can be laborious and time consuming to do manually. In addition, a manual approach might not always be effective in finding certain classes of defects. Test automation offers a possibility to perform these types of testing effectively. Once automated tests have been developed, they can be run quickly and repeatedly. Many times, this can be a cost-effective method for regression testing of software products that have a long maintenance life. Even minor patches over the lifetime of the application can cause existing features to break which were working at an earlier point in time.

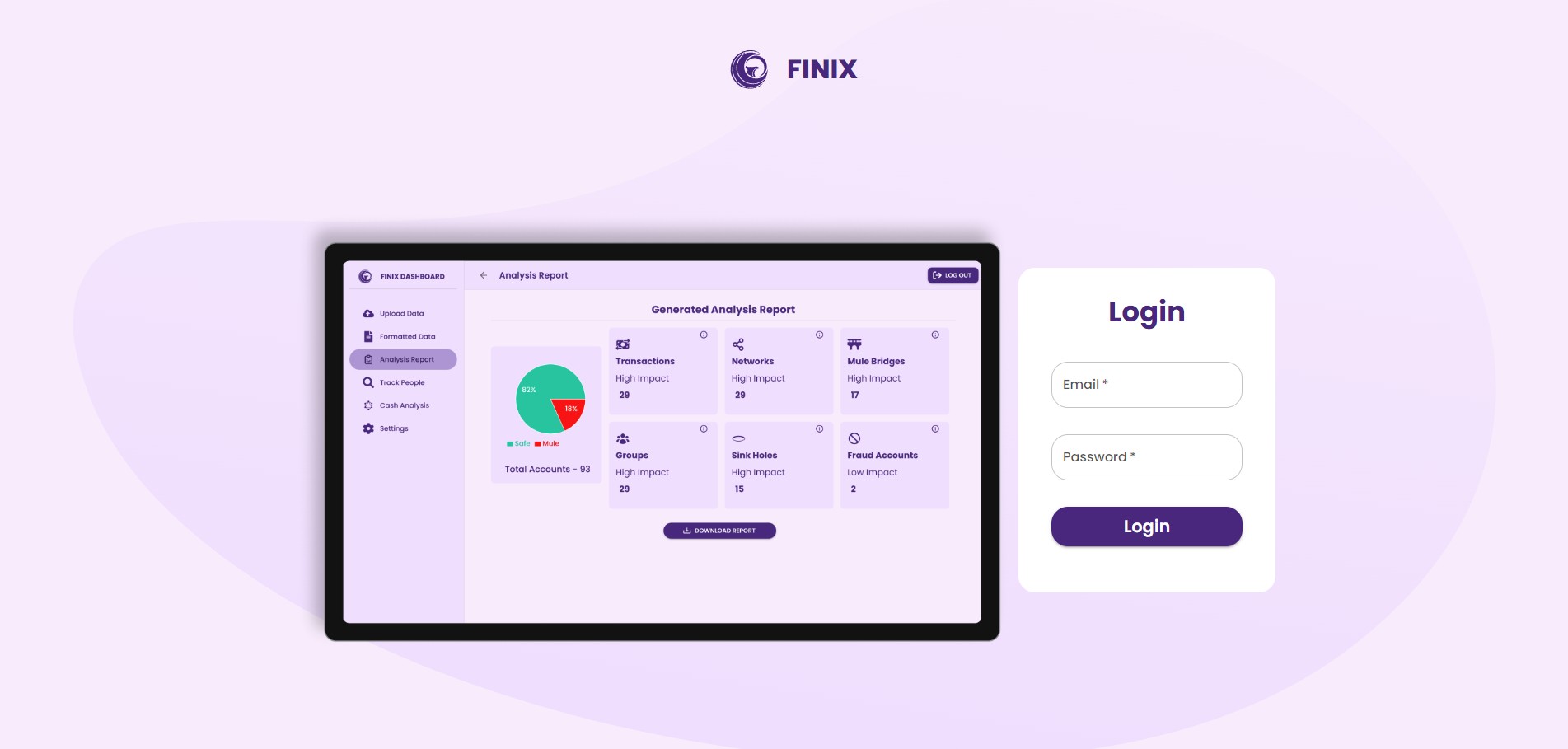
# Activities:

1. Develop a sample web/mobile application.
2. Design software testing methodology in detail for selected application.
3. Explore open source software testing tools for testing web/mobile applications.
4. Document the functionality of testing tools with its features.
5. Document sample application features.
6. Create test cases and test scripts.
7. Run and document test cases.



# Result

**Problem statement-**

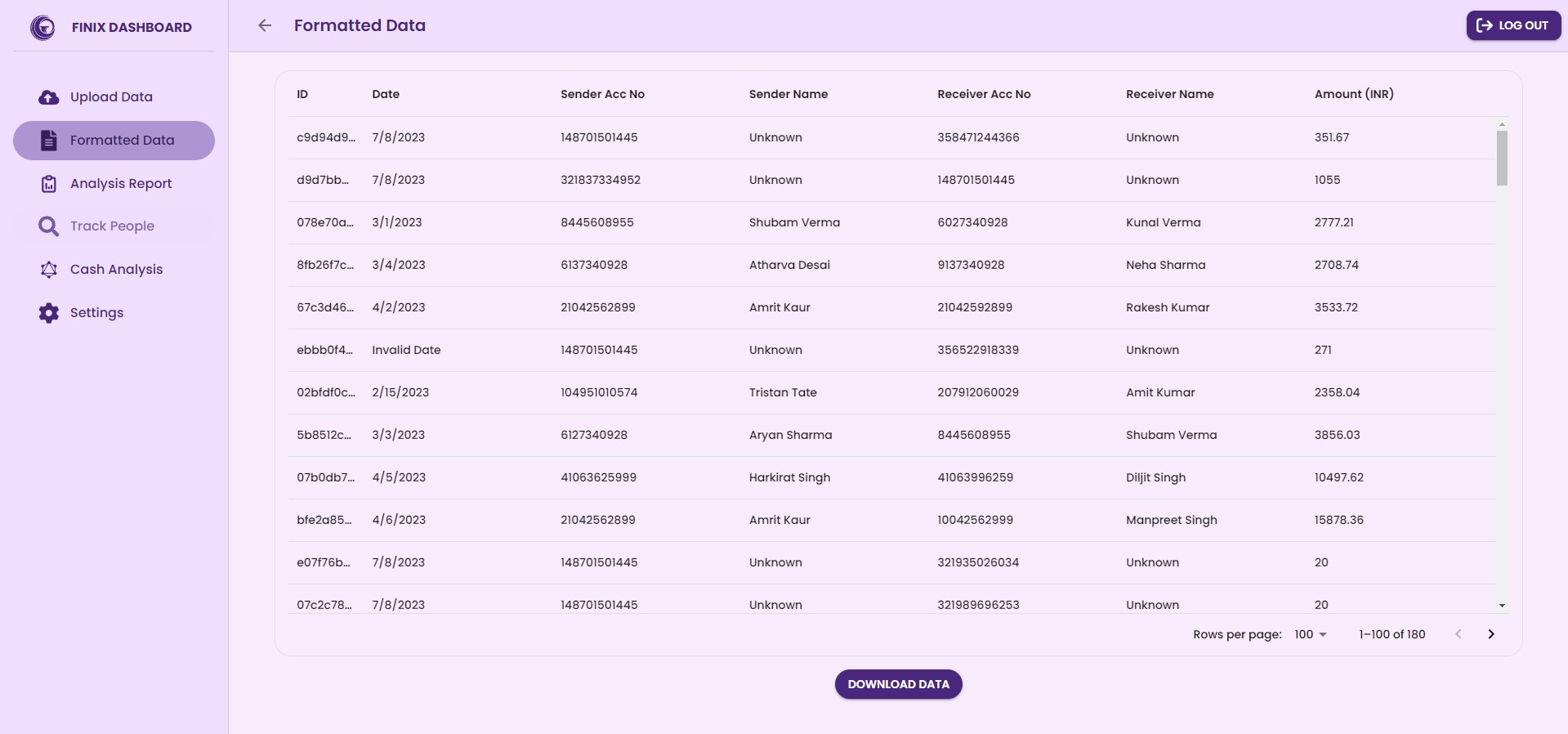
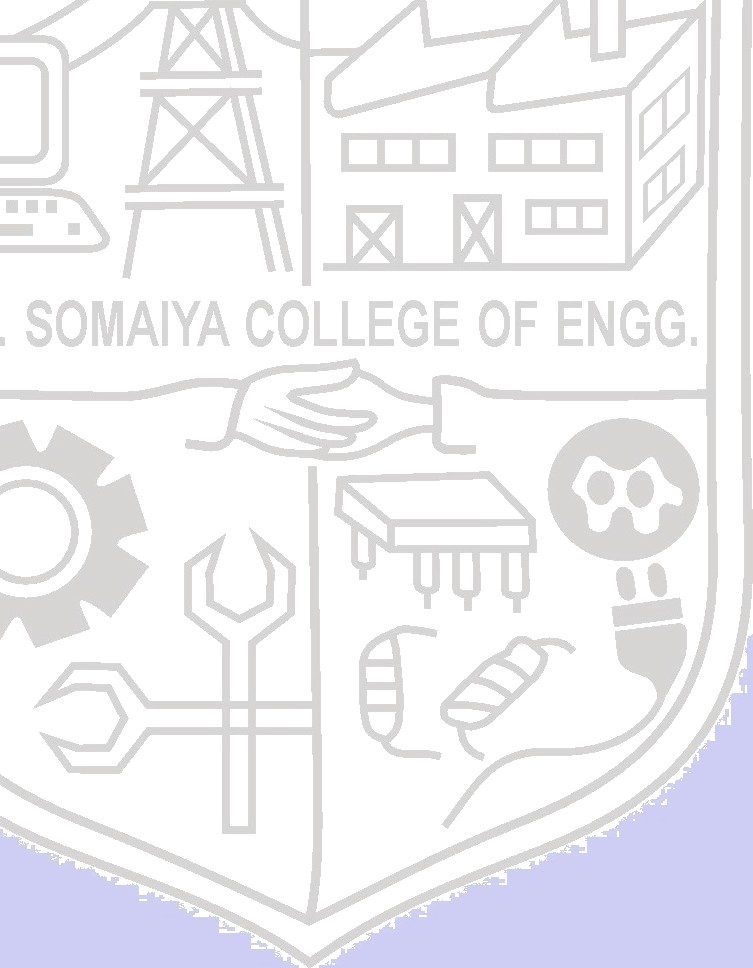
The aim is to design and develop a comprehensive financial trail analysis tool that provides an intuitive and user-friendly interface for users to upload, parse, and analyze financial statements in multiple formats, such as PDFs, CSVs, and Excel files. The platform should also offer advanced features like automated data extraction, a dashboard for visualizing fund flows or systems to track transactions and generate detailed reports. Additionally, it should support audit trails, enabling users to trace fund movements and identify potential discrepancies or compliance issues efficiently.

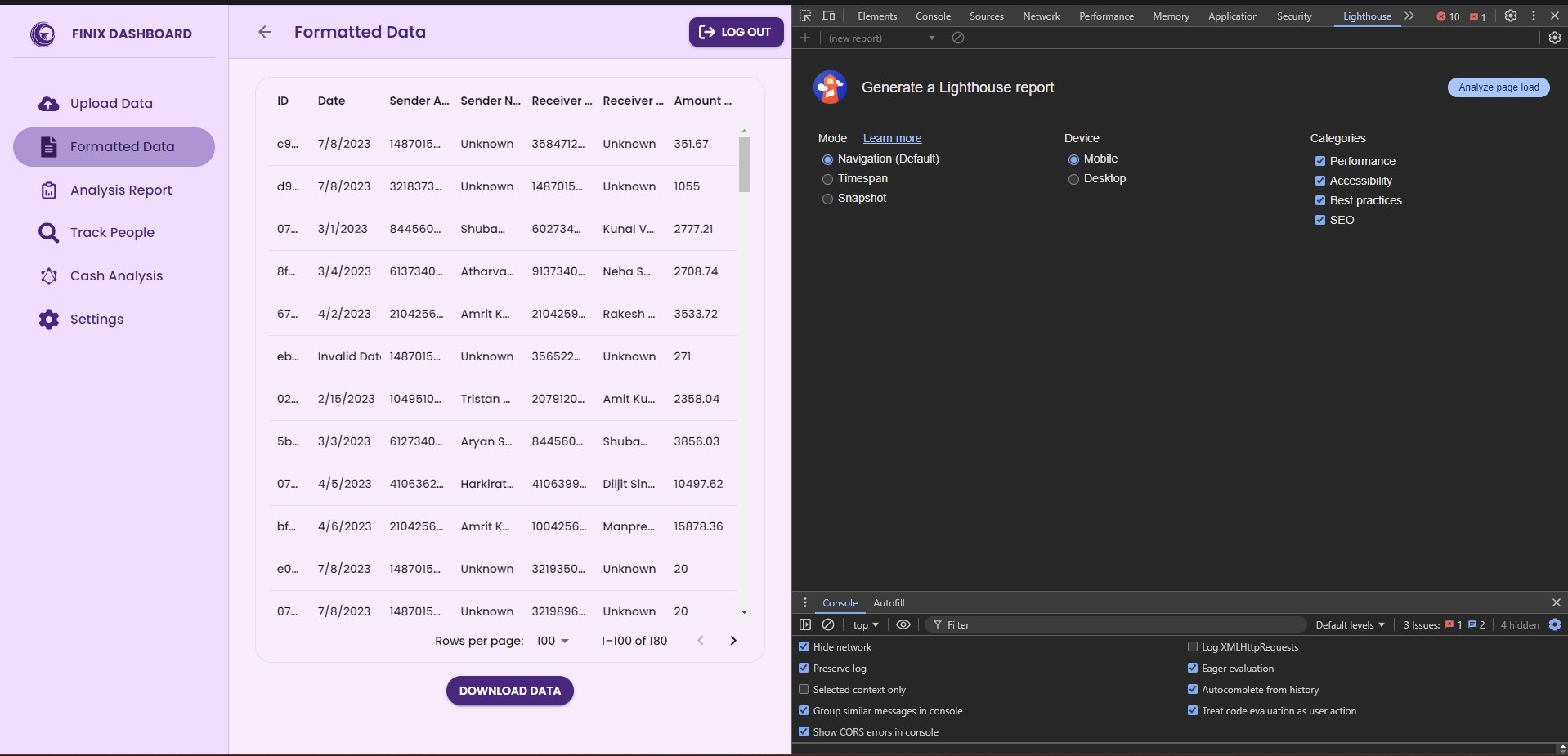
Types of Testing:

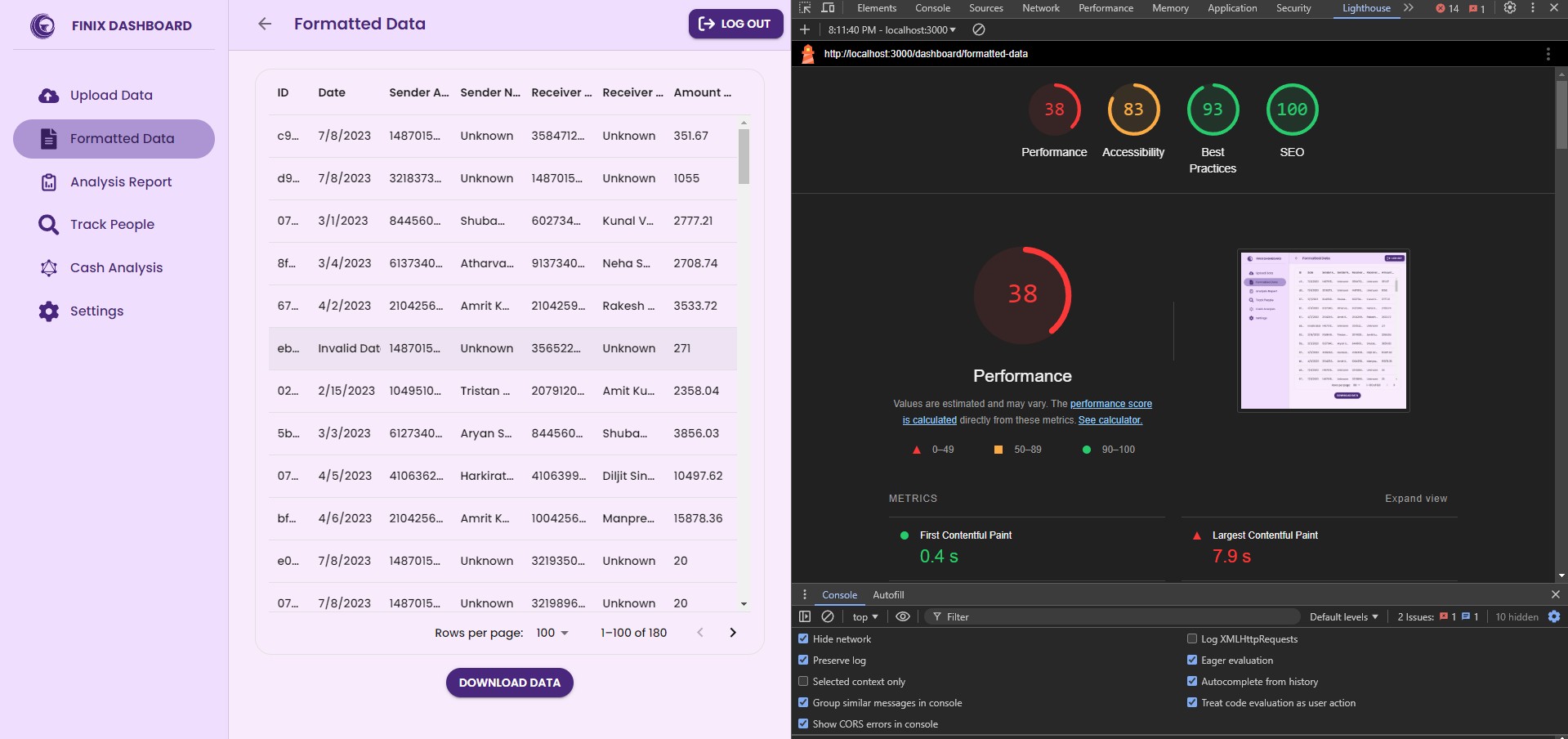
* + Functional Testing: Ensure each feature works as expected.
  + Unit Testing: Test individual components like the file upload and parsing modules.
  + Integration Testing: Ensure components (e.g., the dashboard and data extraction) work well together.
  + System Testing: Test the entire system end-to-end, including file upload to report generation.
  + Performance Testing: Use tools like JMeter to test the system under load.
  + Security Testing: Validate secure access to sensitive data.
  + Usability Testing: Ensure the interface is intuitive.

As the project is currently not hosted we cannot run all kinds of testing. But now we can run

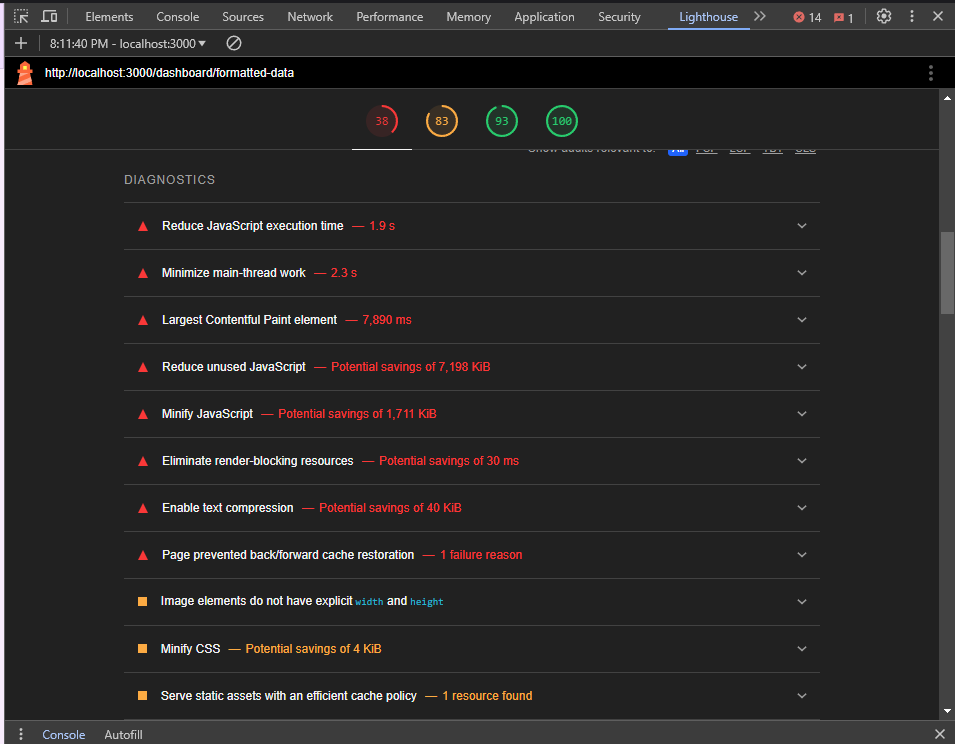
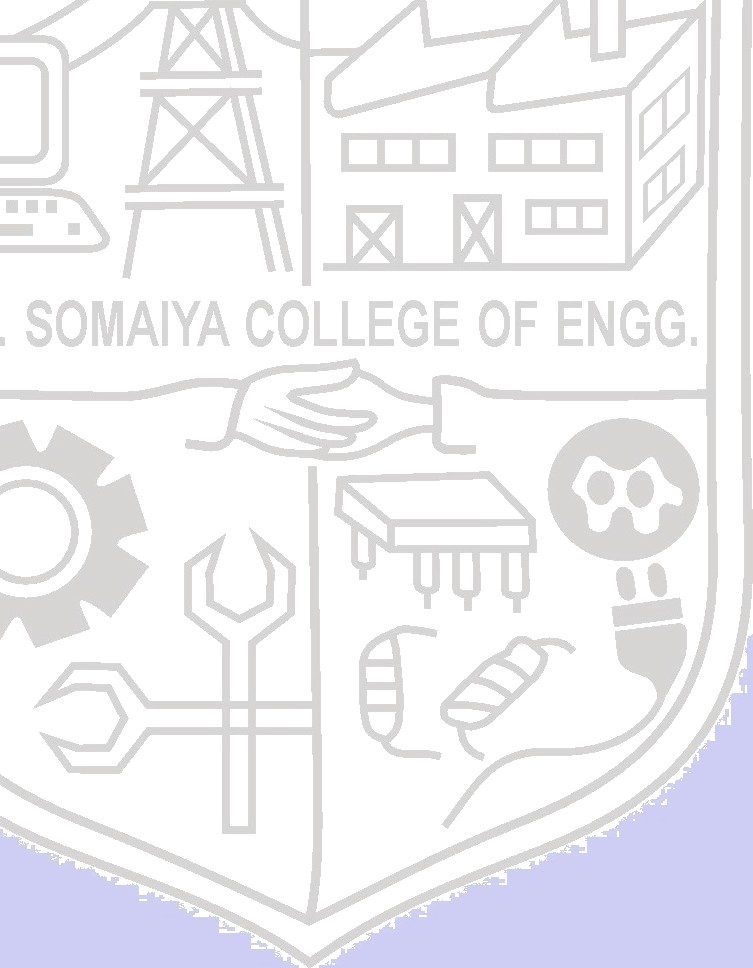
**lighthouse** to see the performance measures



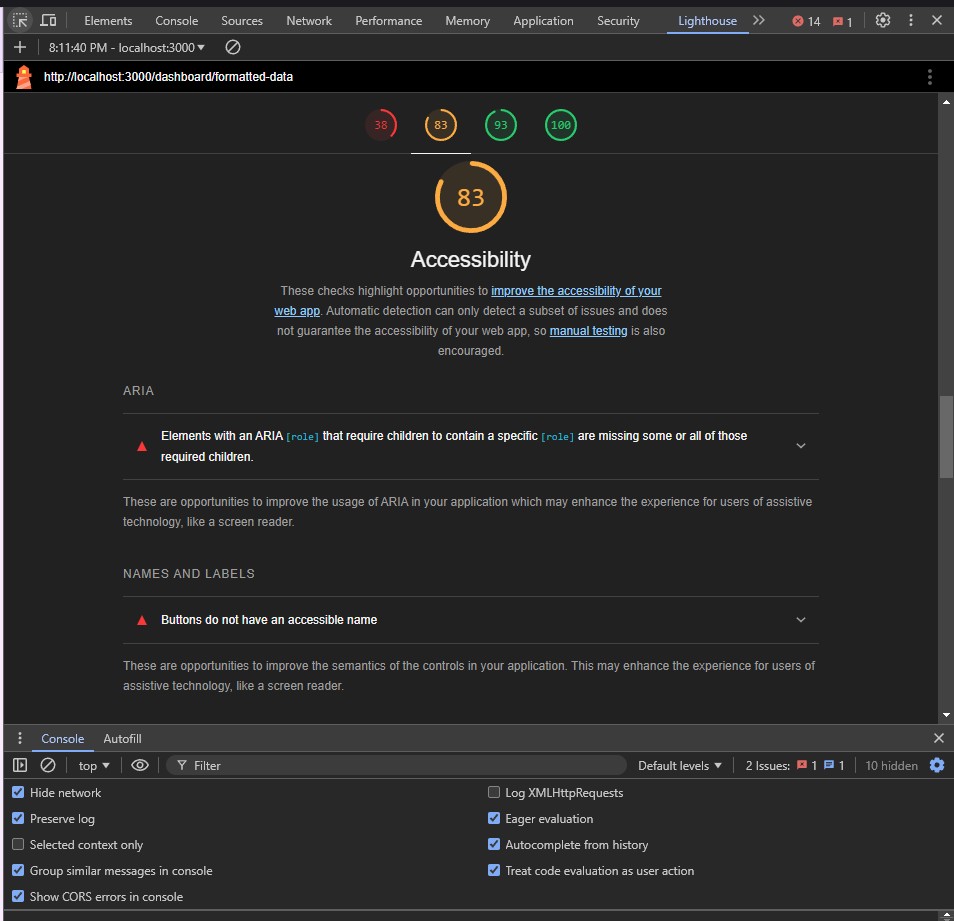
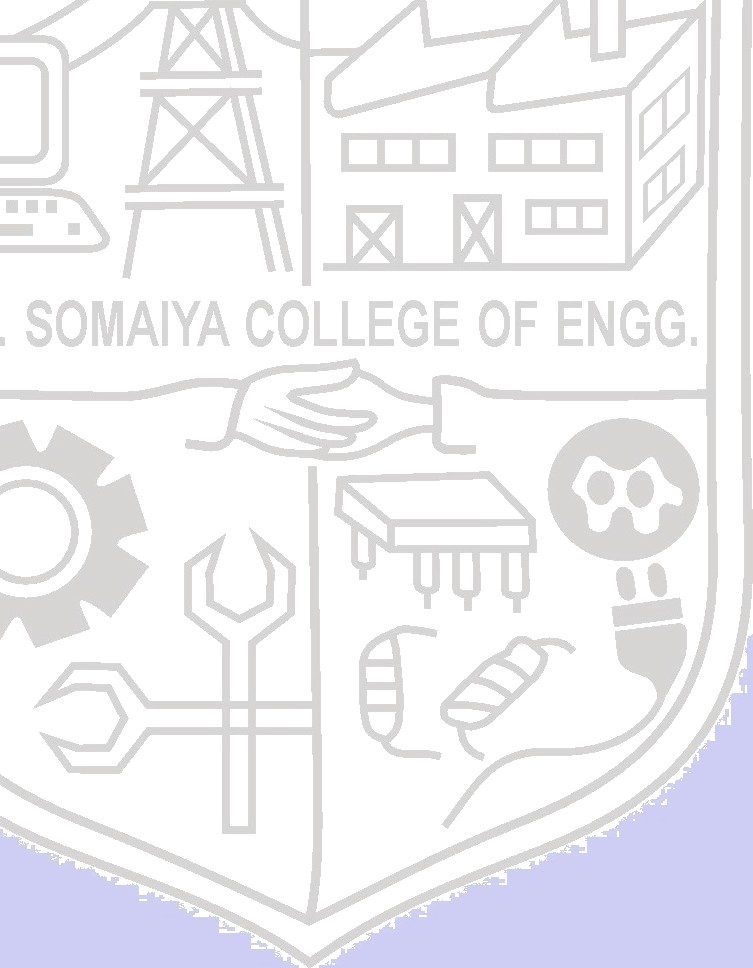




there are lot of performance issues due to heavy frontend libraries which we would work upon in the coming months



Accessibility testing results-

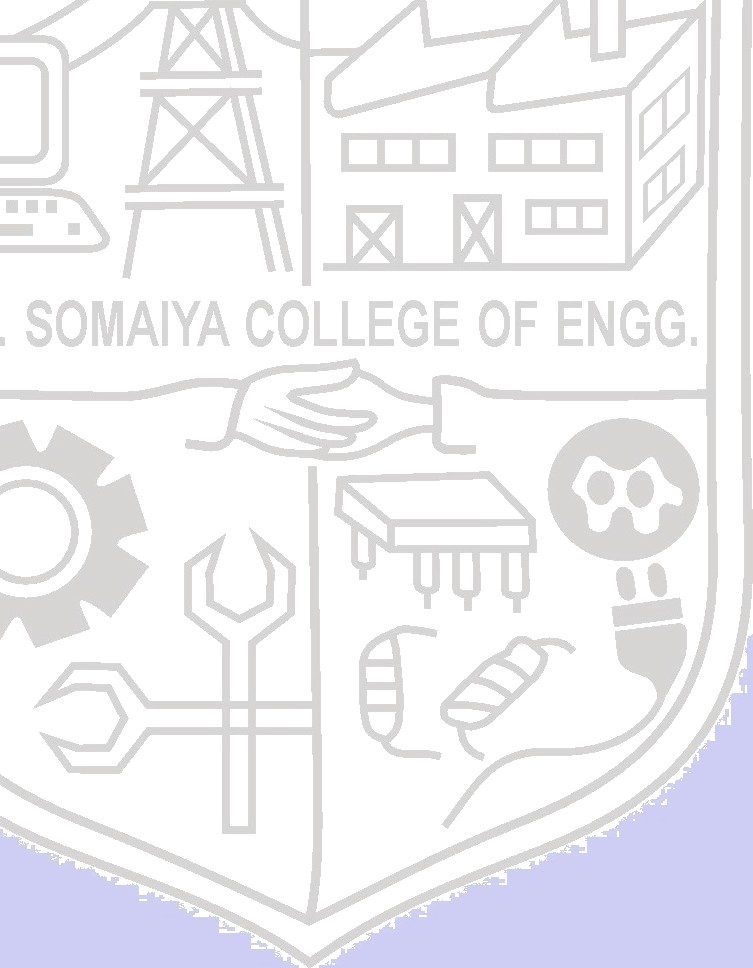


# Questions:

1. Is automation always advantageous? When should one decide to automate test cases? It is not always advantageous to automate test cases. There are some cases like if the task is repeatitive, manual test time is consuming, test is subjected to human error, etc where we need automated testing. But, there are times when manual testing may be more appropriate. For instance, if the application's user interface will change considerably in the near future, then any automation would need to be rewritten. Also, sometimes there simply is not enough time to build test automation. For the short term, Manual testing may be more effective. If an application has a very tight deadline, there is currently no test automation available, and it's imperative that the testing get done within that time frame, the manual testing is the best solution.

# Outcomes:

CO1: Understand software testing concepts and strategies.



CO2: Demonstrate designing and execution of test cases using testing techniques. CO3: Apply recent automation tools for testing software.

CO4: Comprehend different approaches of quality management for software systems

# Conclusion: (Conclusion to be based on outcomes)

Through this project we run various testing on the project

# Grade: AA / AB / BB / BC / CC / CD /DD Signature of faculty in-charge with date

**References:**

# Books/ Journals/ Websites:

1. Software Testing Principles and Practices, Naresh Chauhan, Second Edition, Oxford Higher Education
2. Practical Software Testing, Ilene Burnstein, First Edition, Springer International Edition
3. Effective Methods for Software Testing, Third edition by Willam E. Perry, Wiley Publication
4. <http://www.softwaretestinghelp.com/most-popular-web-application-testing-tools/>
5. <http://www.softwaretestinghelp.com/best-mobile-testing-tools/>
6. <http://www.seleniumhq.org/>
7. <http://appium.io/>