Secure Authentication System Development

Department of Data Science

Information Security (CS3002)

Group Members:

Muhammad Shamil Umar (21i-1786)

Hamiz Ahmed Siddiqui (21i-1678)

Afnaan Asif (21i-1387)

Contents

Chapter 1

Objective

This report aims to display all the workings and findings done for iteration 3 of our project. Initially we planned to work on backend of our application but due to changes in requirement, we decided to adjust our iteration based on new requirements. New requirements stated that we are required to provide detailed quantitative testing report for core functionalities of our project. Hence we decided to temporarily focus more on fulfilling main functional requirement of project which is providing functionality of One Time Password (OTP) authentication. As it is most critical role of our application, it became main focal point of our iteration. We aimed to fully implement OTP process and ensure robustness through extensive quantitative testing. This decision allowed us to allocate more time and resources towards testing and validating the OTP system, ultimately contributing to the overall stability and security of the application.

Chapter 2

Testing

2.1 Features to Test

- **OTP Authentication:**

  We implemented secure generation of OTP using python modules and used SMTP module to send OTP to users.

- **Unique OTP generation:**

  We had to make sure that each user receives different password so it doesn't effect any other user.

- **Frontend Integration:**

  We connected frontend react application to our API endpoints to provide complete functionality of sign up feature.

- **Database Integration:**

  We established and maintained connection with our database to make sure each user who used our application to sign up has his/her data stored in our database.

2.2 Testing Methods

1. **Email Delivery Time:** We measured time from user requesting the OTP to the time it took user to get that OTP.

2. **Load Handling:** tested how many queries can our system handle. As our application

is not deployed, we had to run separate script which would continuously request for OTP.

3. **Database Query Time:** We measured time it took for user data to be registered in our database once received OTP is entered by user.

2.3 Results

Test	Results
Email Delivery Time	3-5 seconds
Load Handling	70-73% success rate
Database Query Time	<1 second

Table 2.1: Results of tests mentioned in section 2.2

2.3.1 Test 1:

For this test we wrote script that would make 20 api queries per script using python script. We utilized port forwarding to share application access to all 3 members. On average it took 3-5 seconds for user to receive emails with OTP. Major portion is dependent of how much time it takes by SMTP module to send email to user over the internet, which we can not control.

2.3.2 Test 2:

For this test we wrote python script that would bombard our system with request. Each script generated 100 queries to API endpoints and around 70% of them responded with success while others threw error probably because system can not handle all the requests.

2.3.3 Test 3:

We couldn't write script script for it so we manually tested it and it took probably less then a second to write data to database. Either we access this application with host system our any other system, in both cases we got same results.