Syeda Reeha Quasar

14114802719

4C7

Aim

To perform various testing using the testing tool unit testing, integration testing for a sample code of the suggested system.

Experiment - 10

Software Engineering Lab

# **EXPERIMENT – 10**

## **Aim:**

To perform various testing using the testing tool unit testing, integration testing for a sample code of the suggested system.

## **Theory:**

“Testing is the process of executing a program with the intent of finding errors.”

The purpose of software testing is:

1. To demonstrate that the product performs each function intended.
2. To demonstrate that the internal operation of the product performs according to specification and all internal components have been adequately exercised.
3. To increase our confidence in the proper functioning of the software.
4. To show the product is free from defect.
5. All of the above.

**Unit Testing —** Checks each coded module for the presence of bugs. Unit testing’s purpose is to ensure that each as- built module behaves according to its specification defined during detailed design.

**Integration Testing ----** Interconnects sets of previously tested modules to ensure that the sets behave as well as they did as independently tested modules. Integration testing’s purpose is to ensure that each as-built component behaves according to its specification defined during preliminary design.

## **Performance Instruction:**

**Unit test planning—**

Generate plans and procedures to test each module independently and thoroughly.

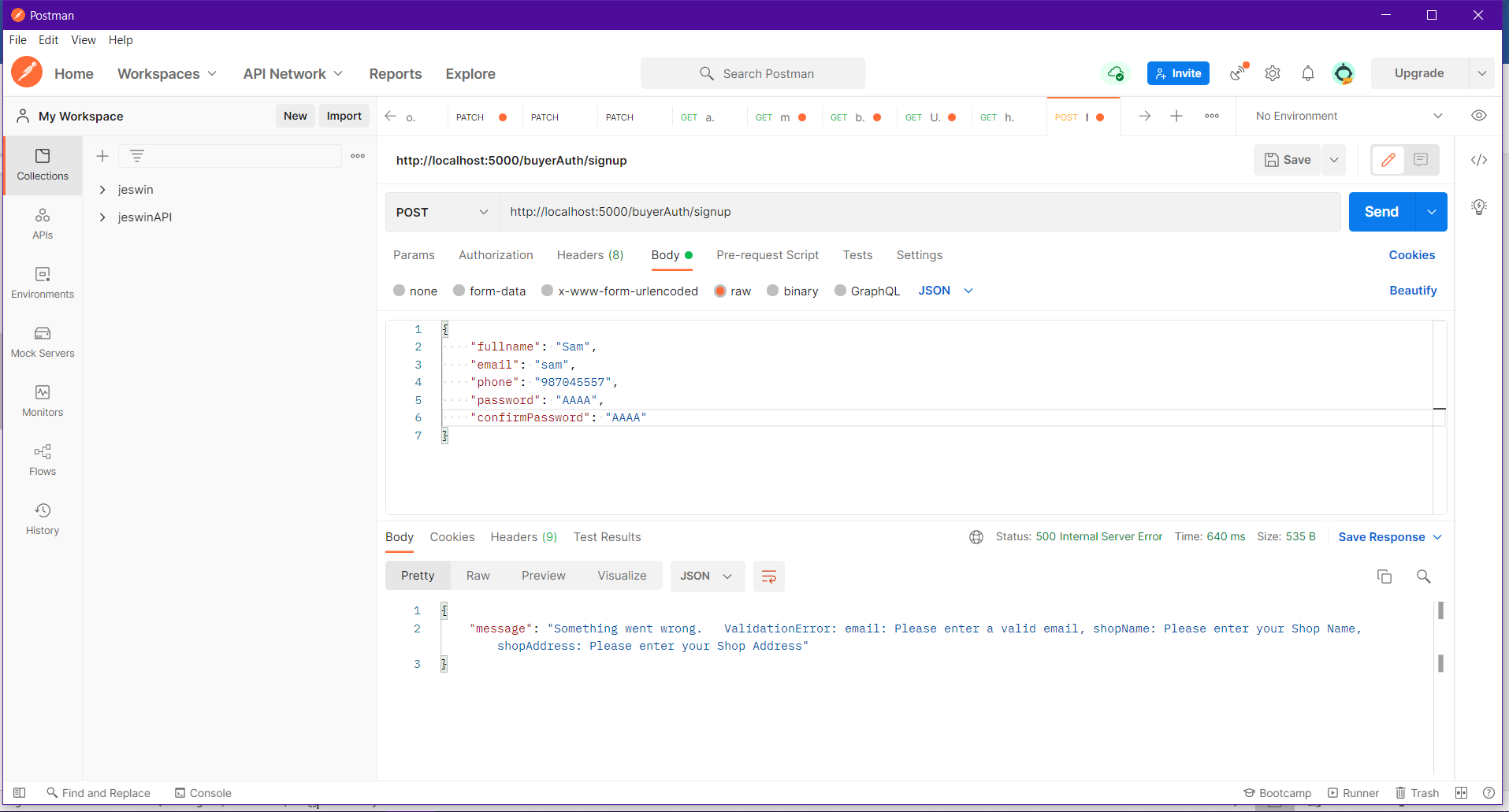
**Integration Test Planning—**

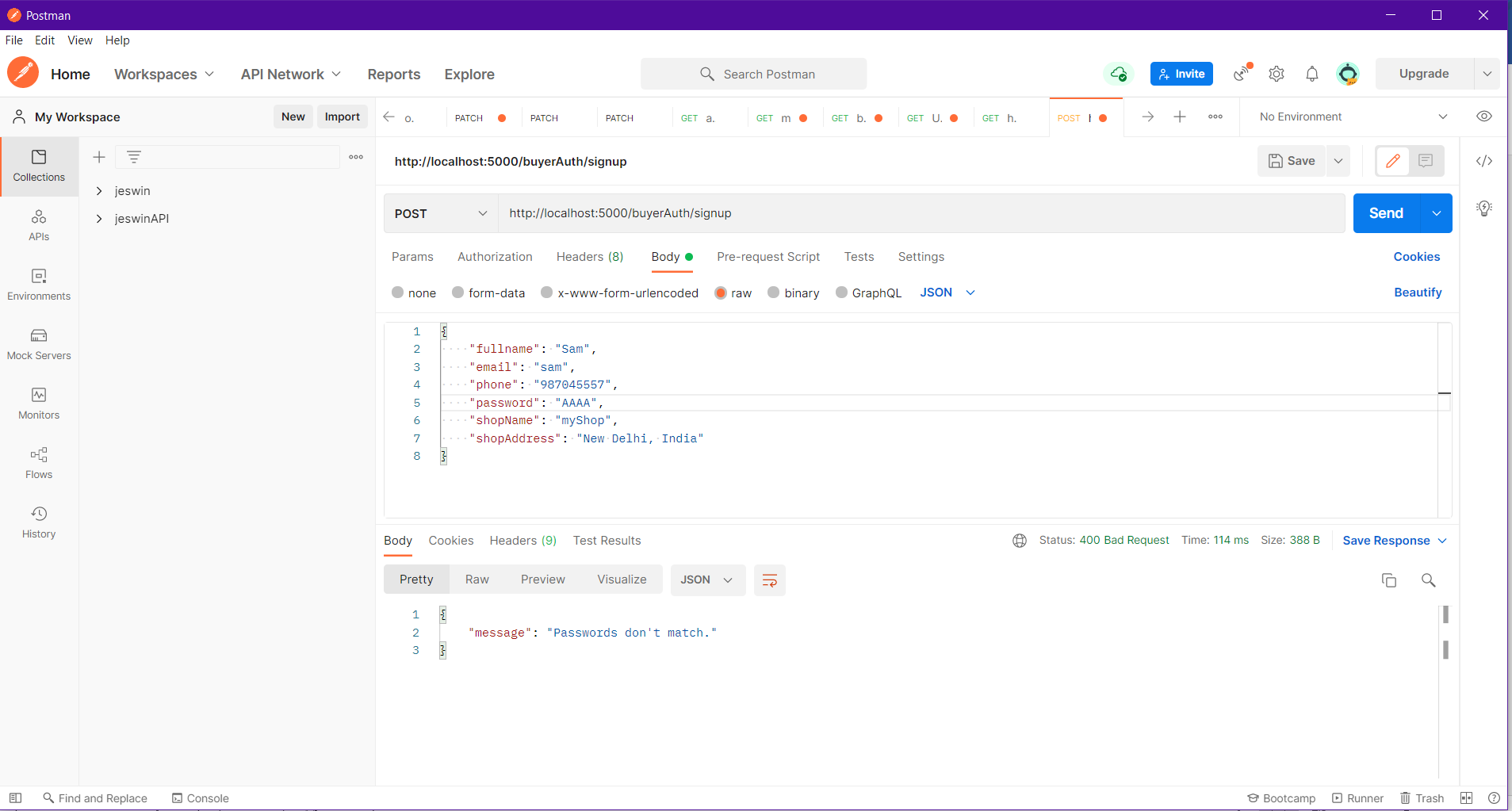
Generate plans and procedures to effect orderly system integration.

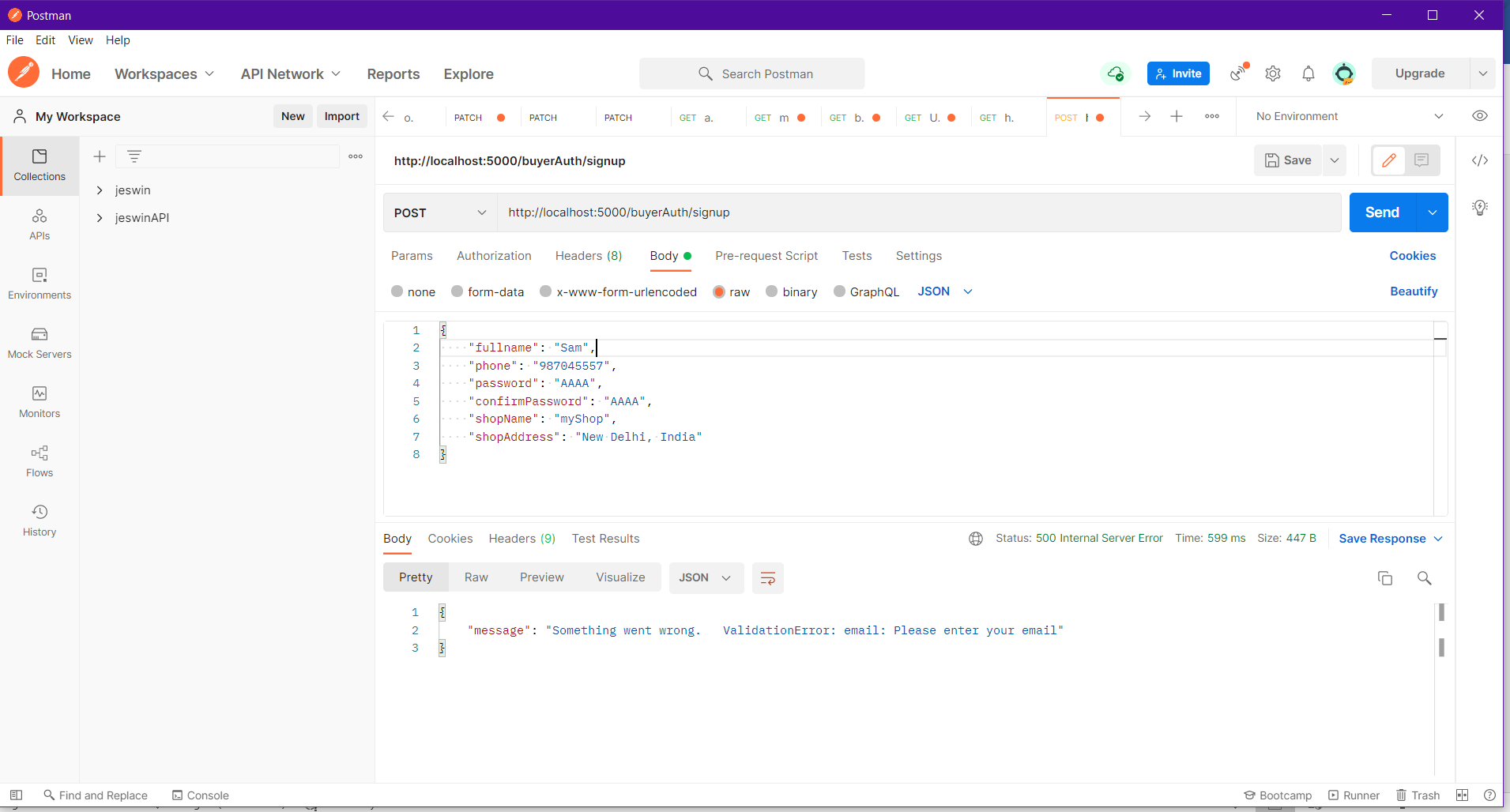
## **Output:**

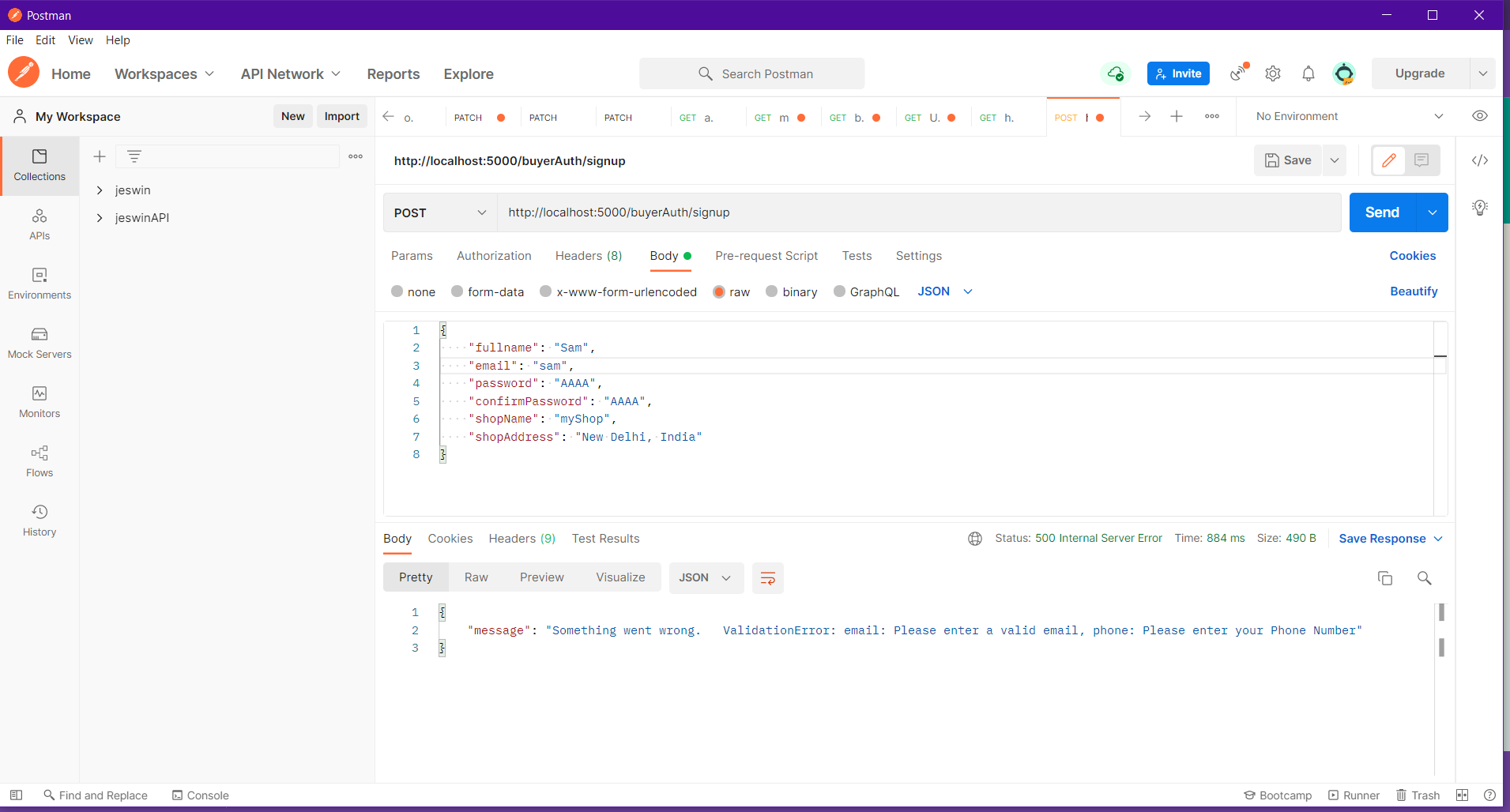
**Backend validations: - Authentication module -> (same for buyer, seller and deliverer)**

**Missing fields – “Something went wrong”**



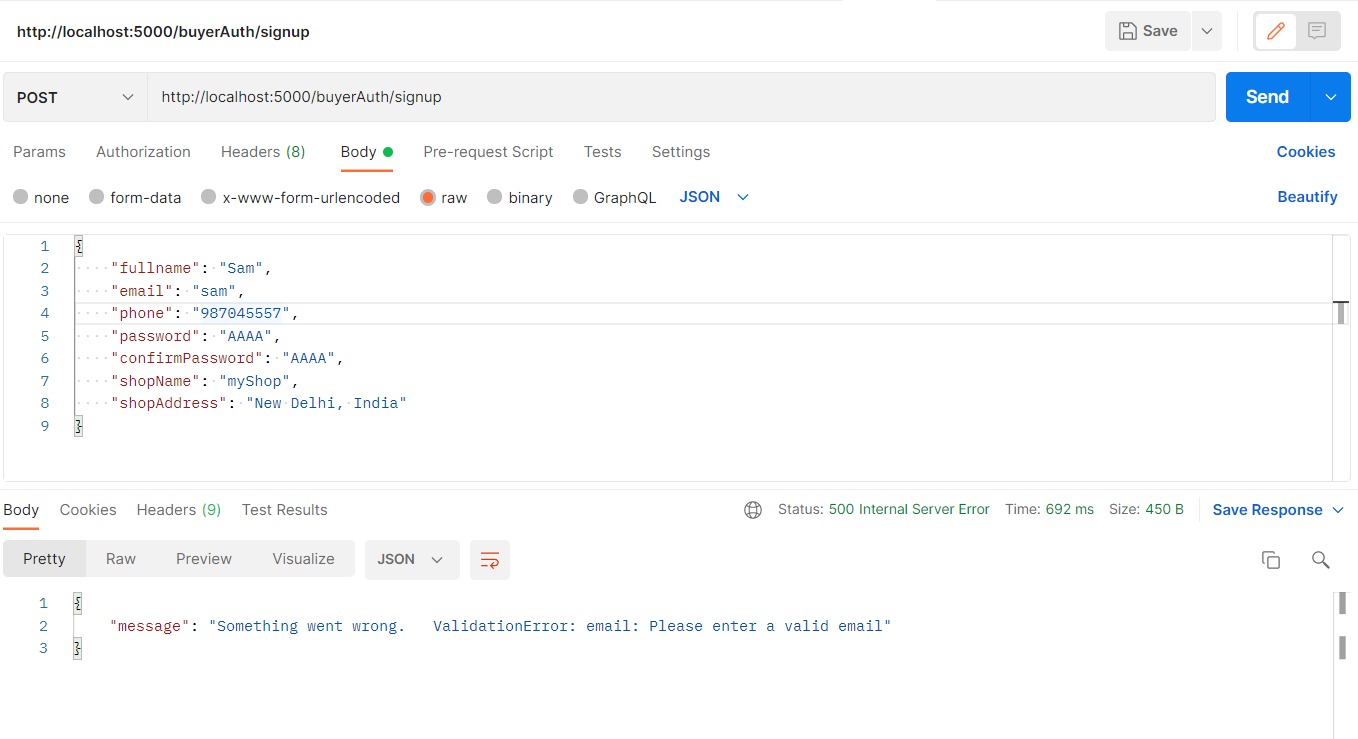




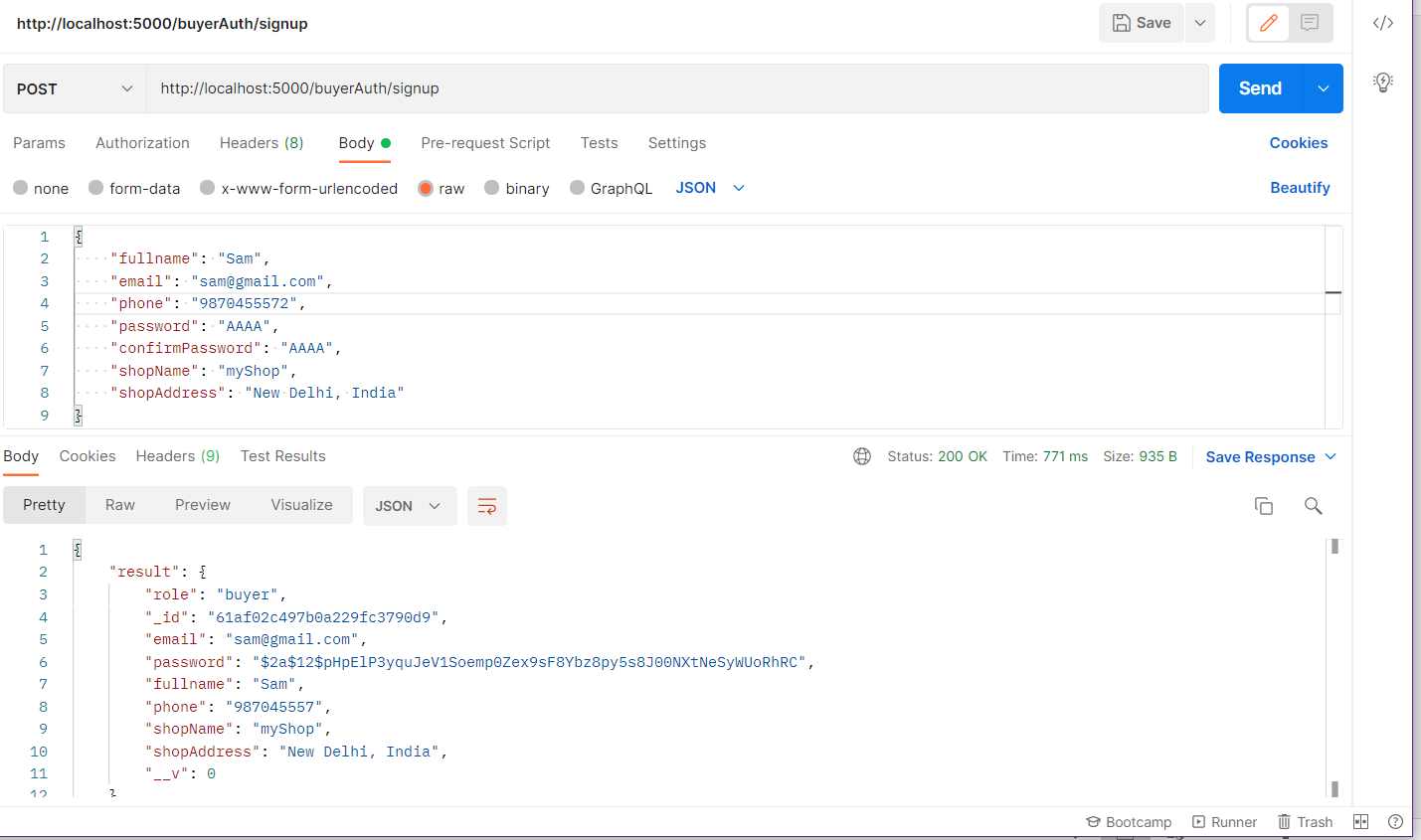


**Not a valid email or phone**

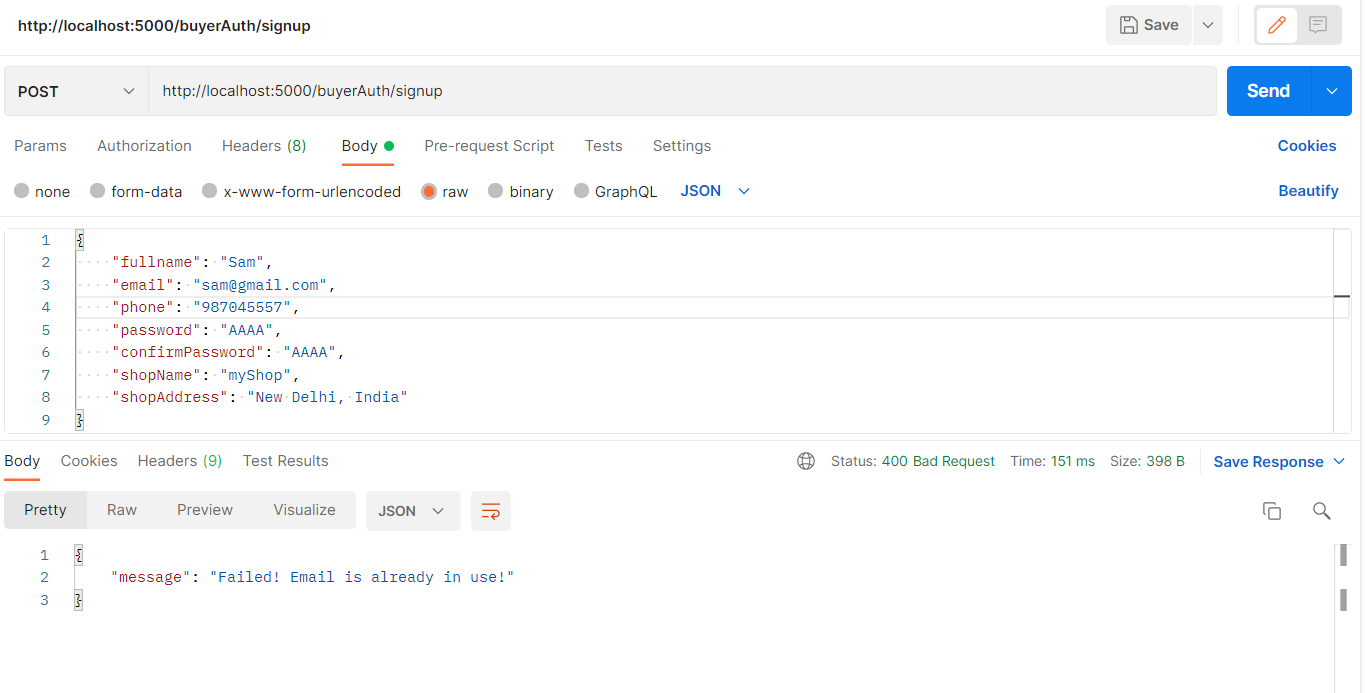
**“Validation errors”**

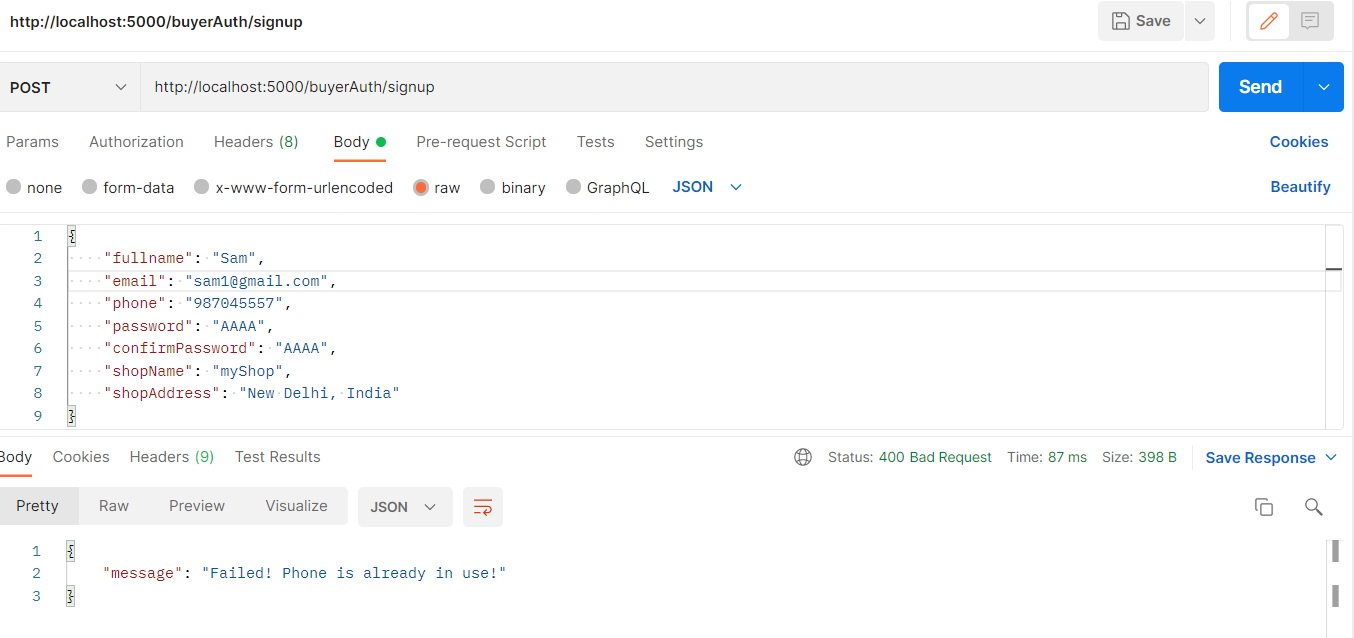


**When everything is valid user get’s signed up**

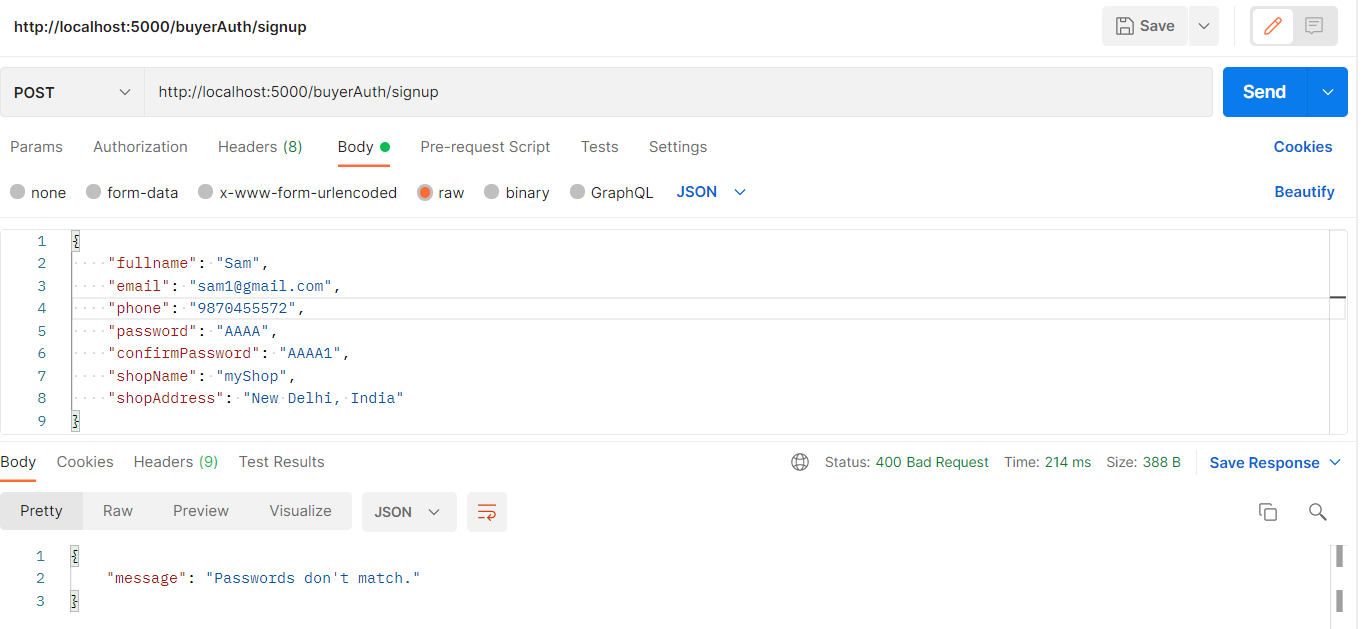


**If current email and phone number already in use!**

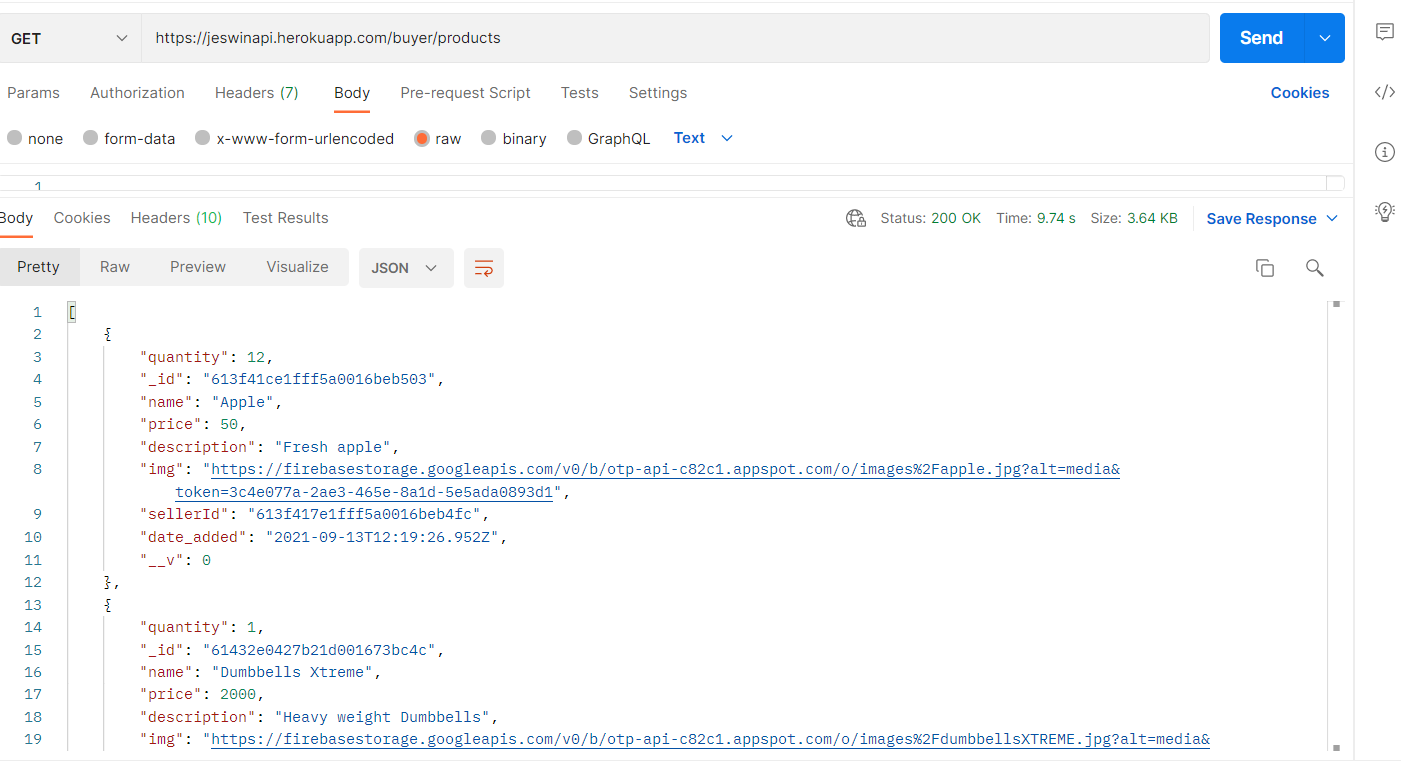




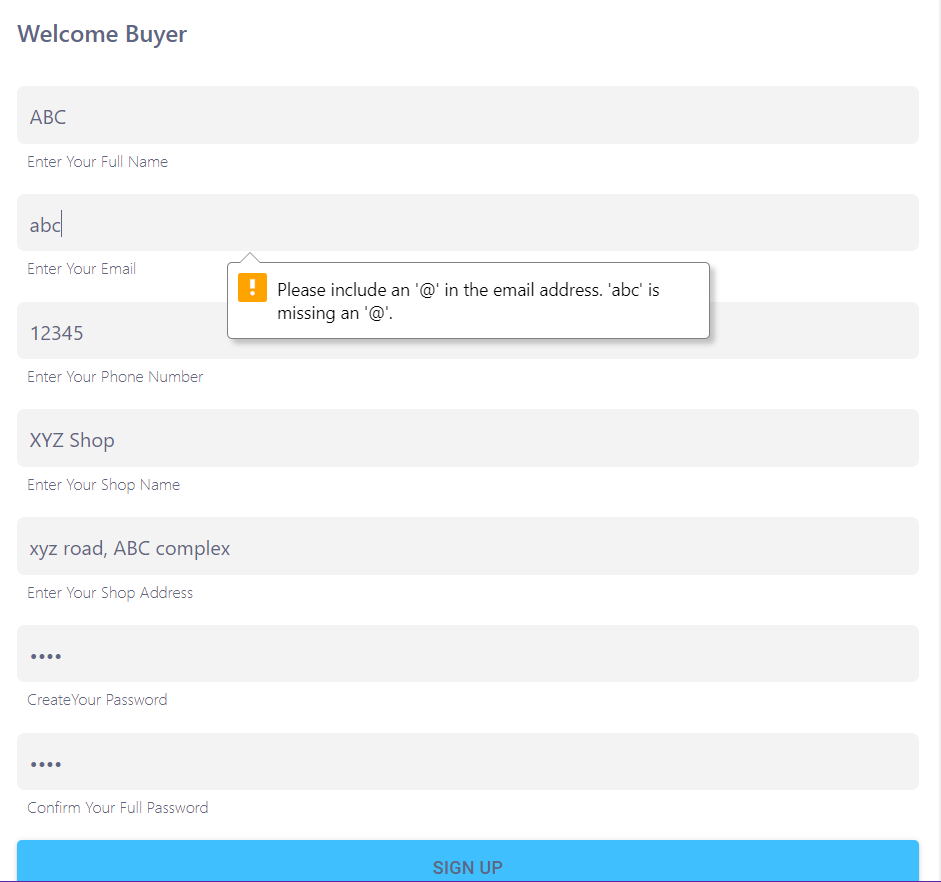
**Password doesn’t match error**

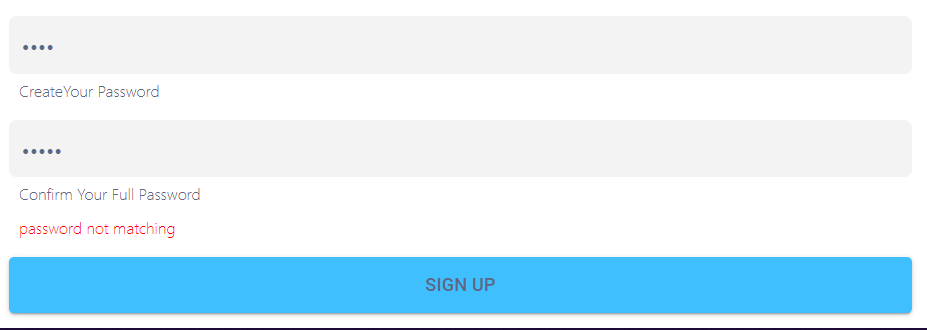


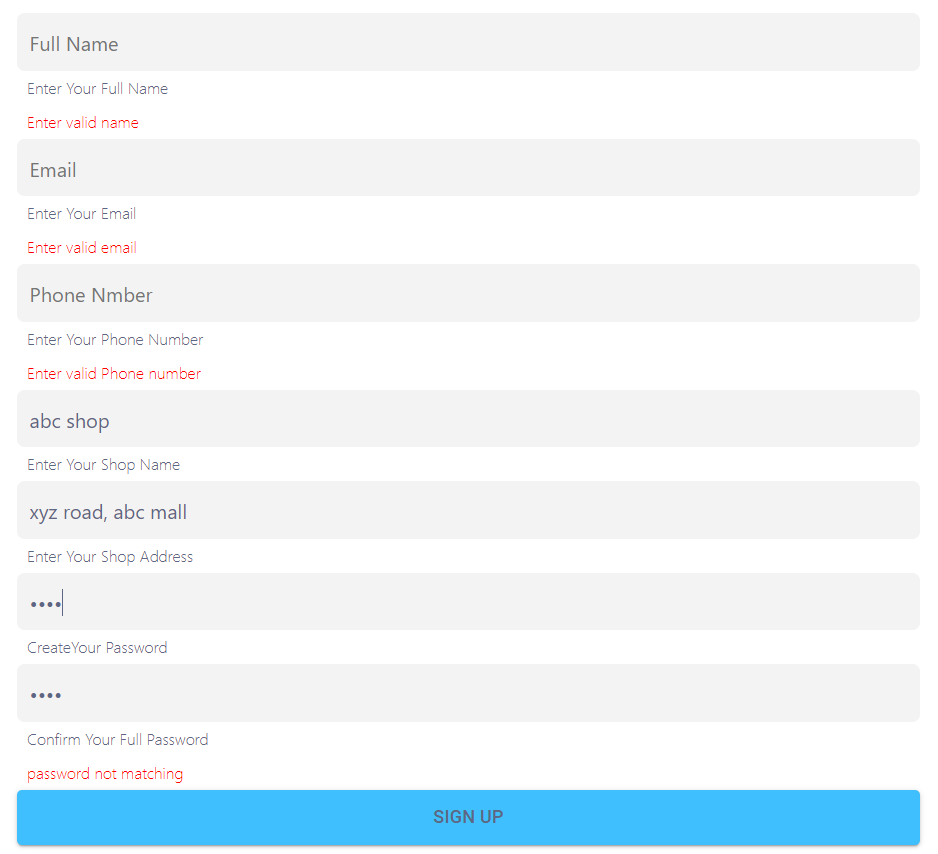
Similarly for all APIs there is different error msgs in string and is displayed to cover edge cases



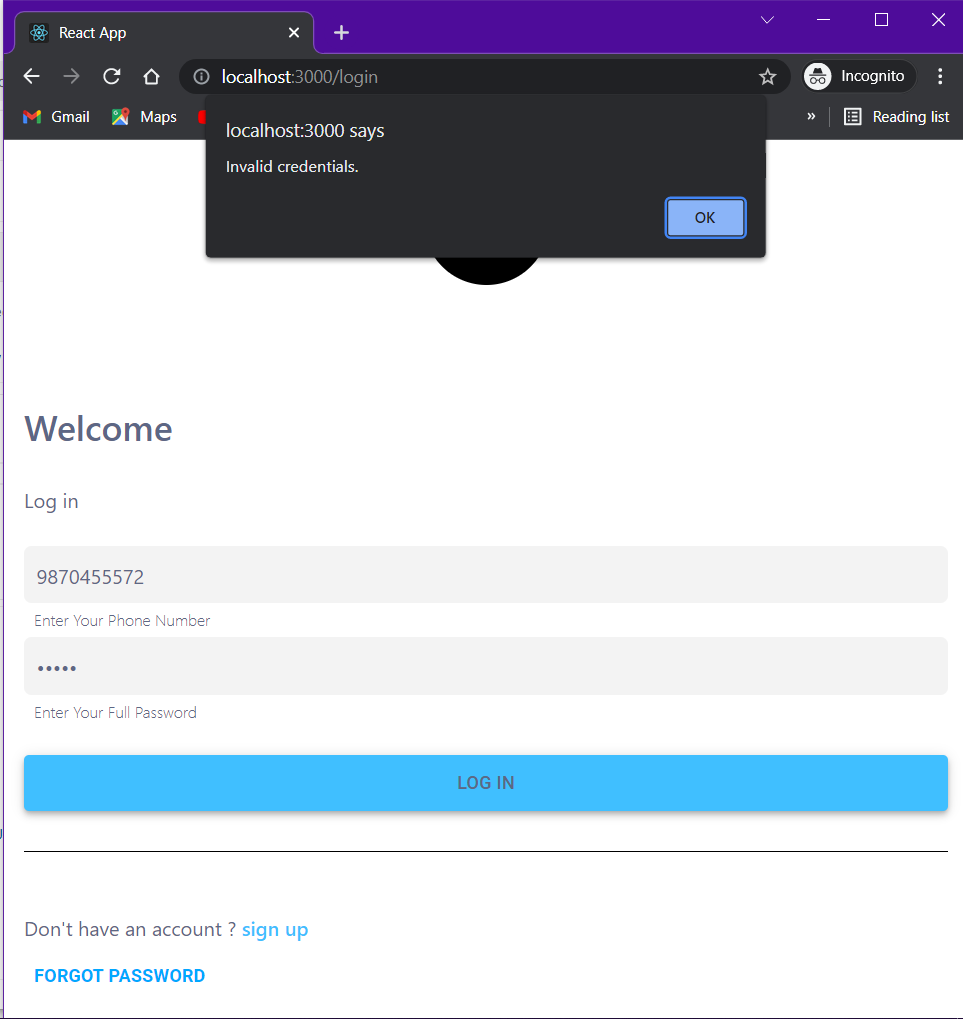
**Frontend validations: - Authentication module -> (same for buyer, seller and deliverer)**

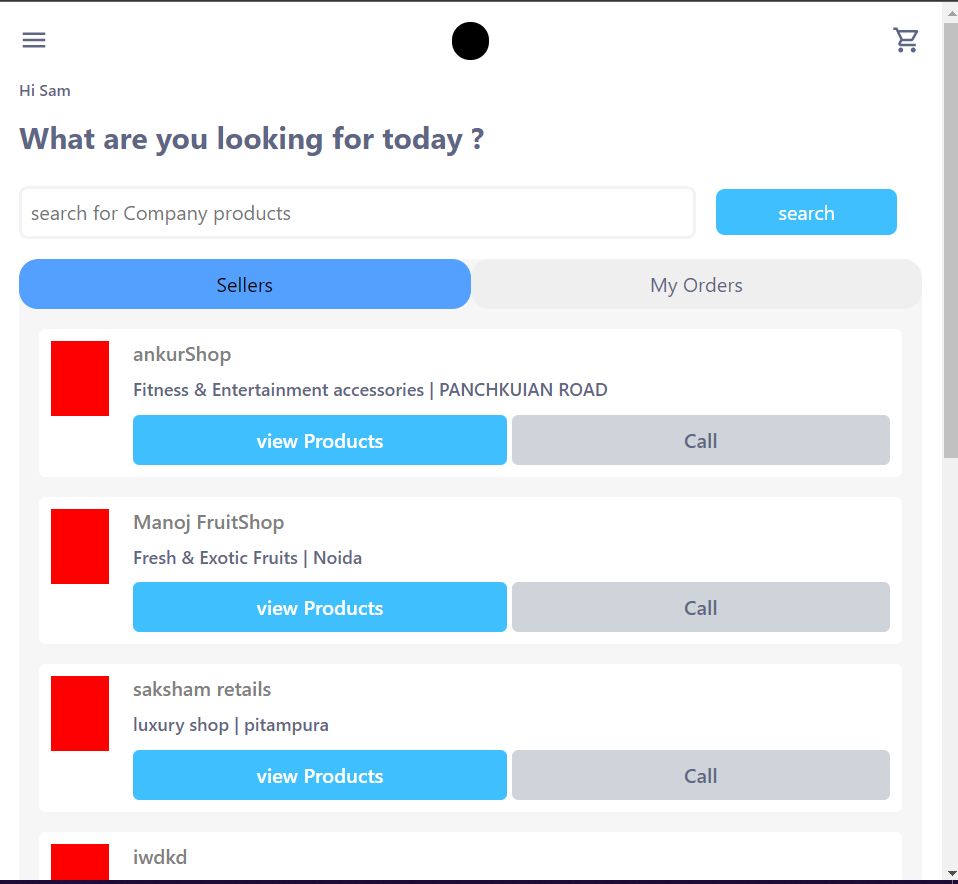
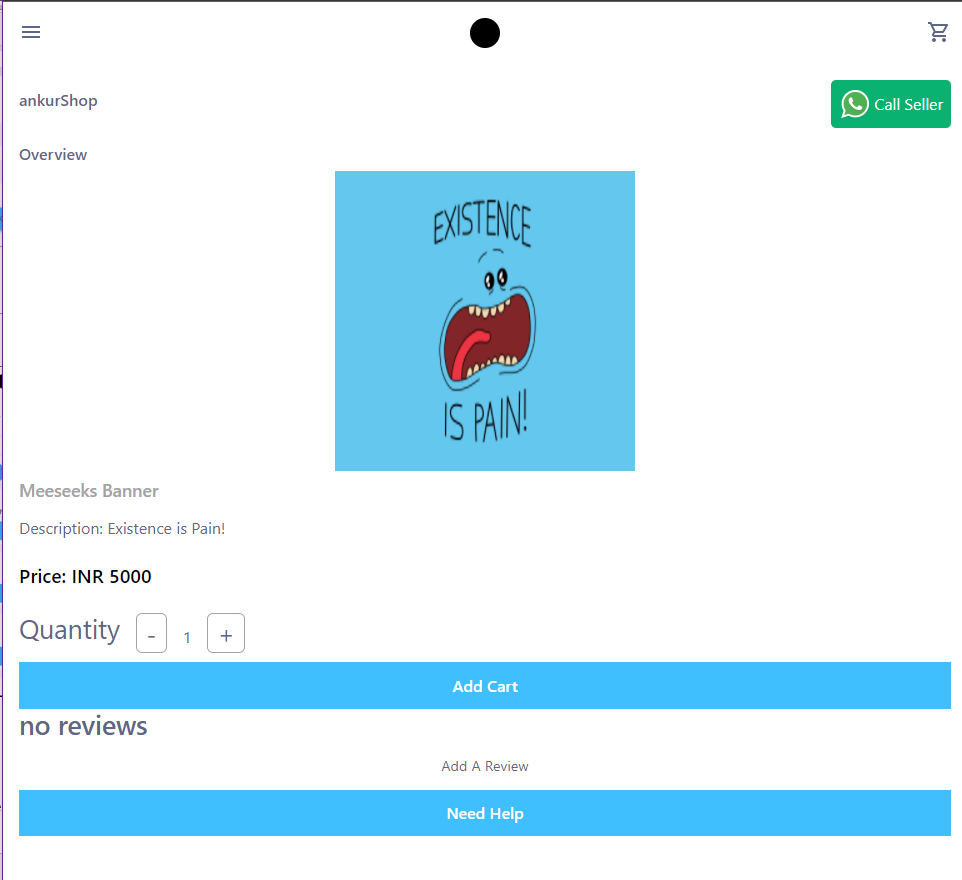


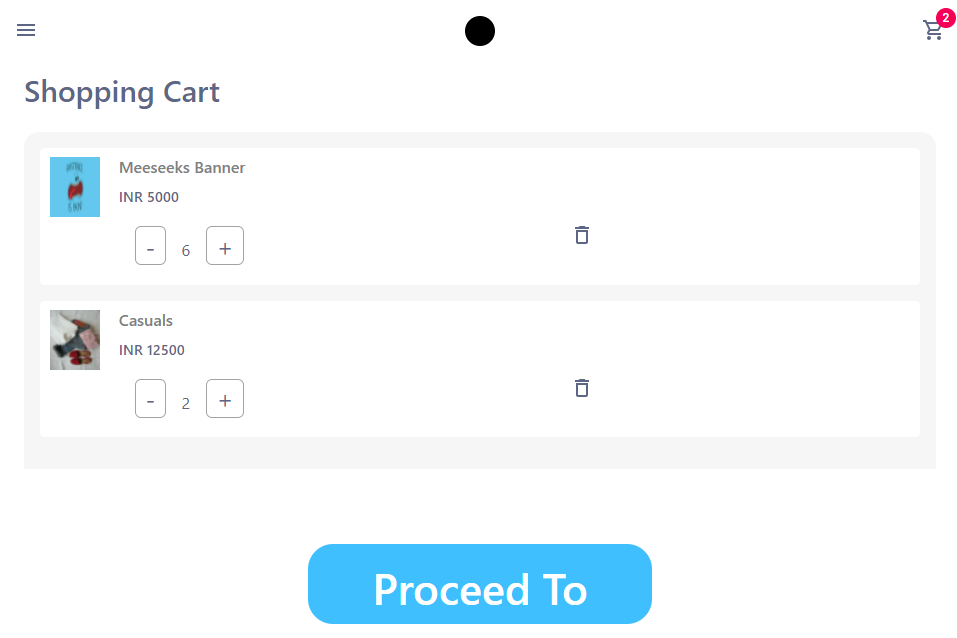
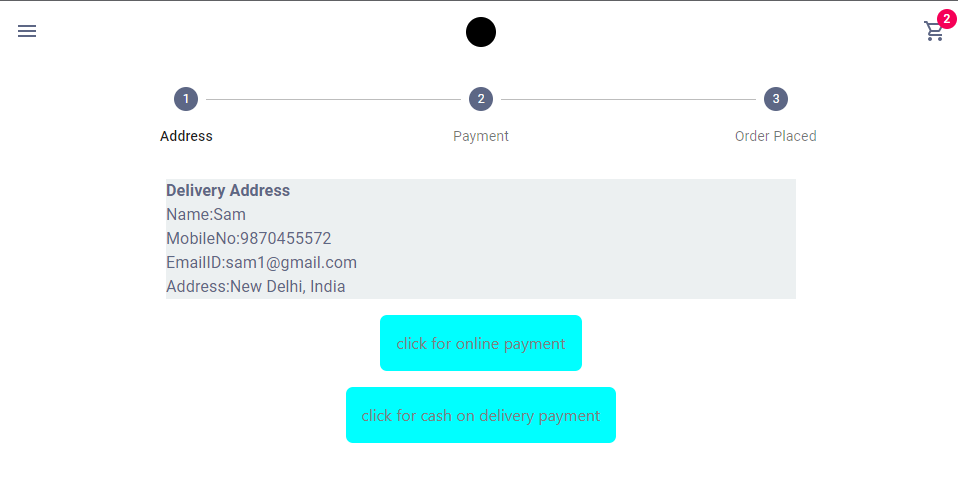


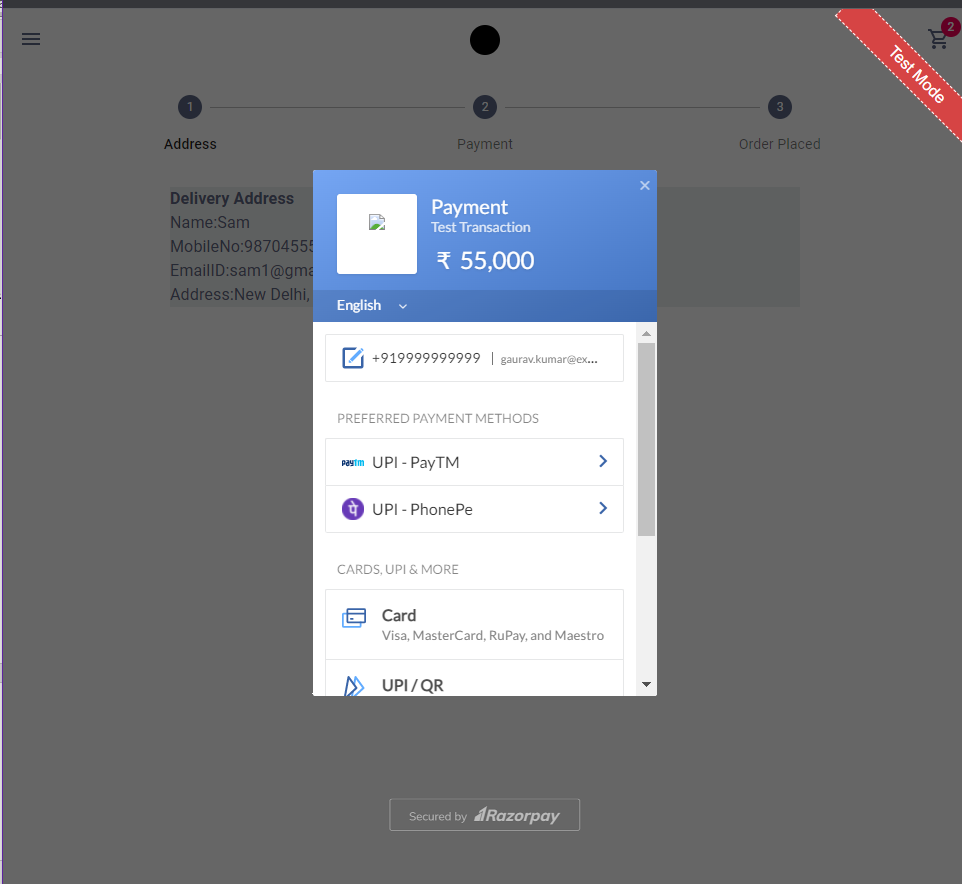
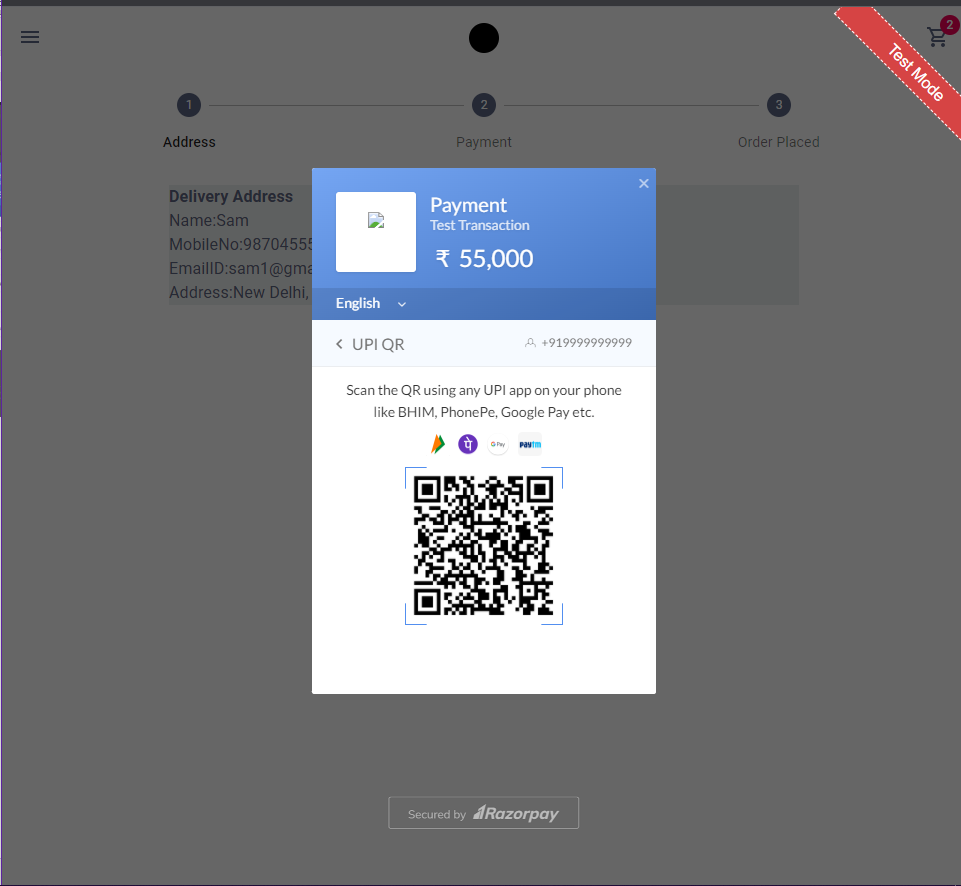


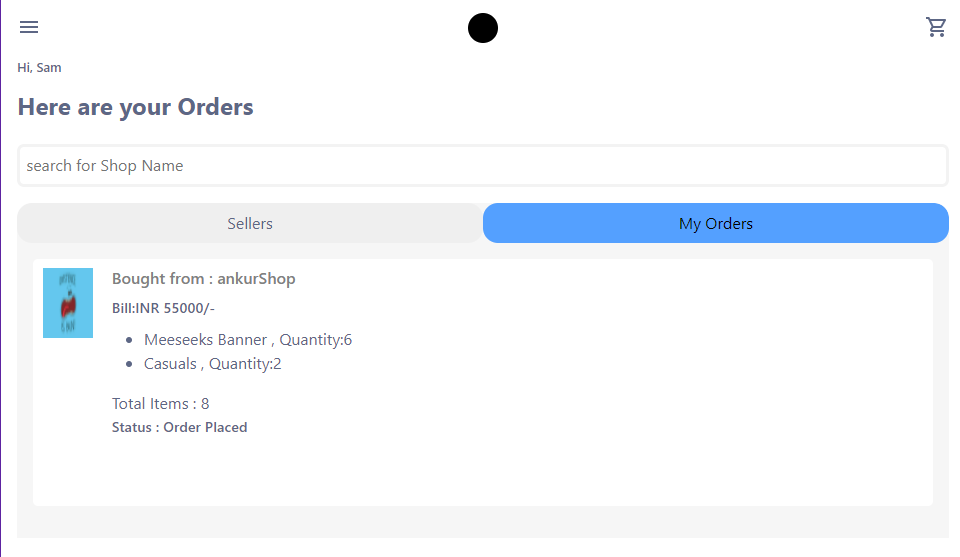
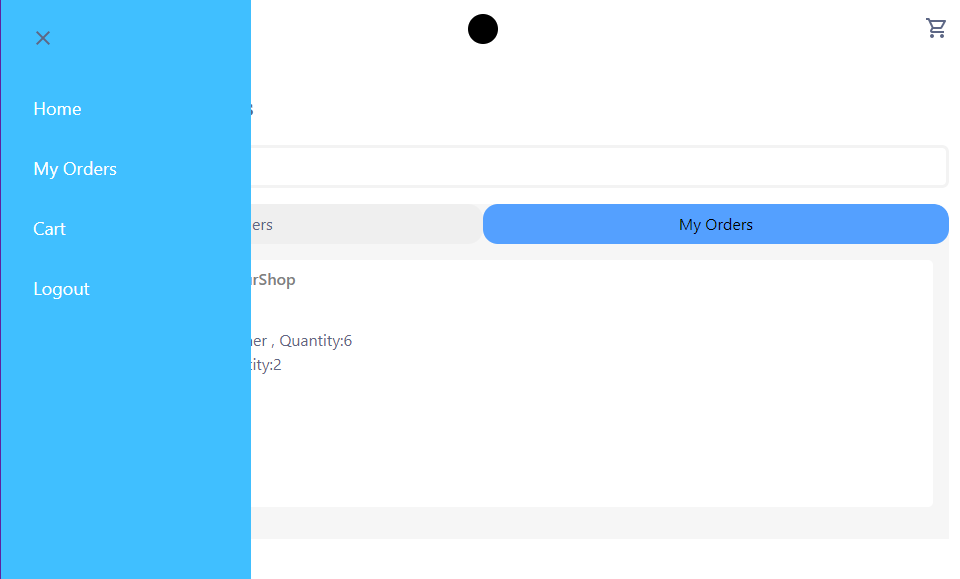
Invalid Credentials

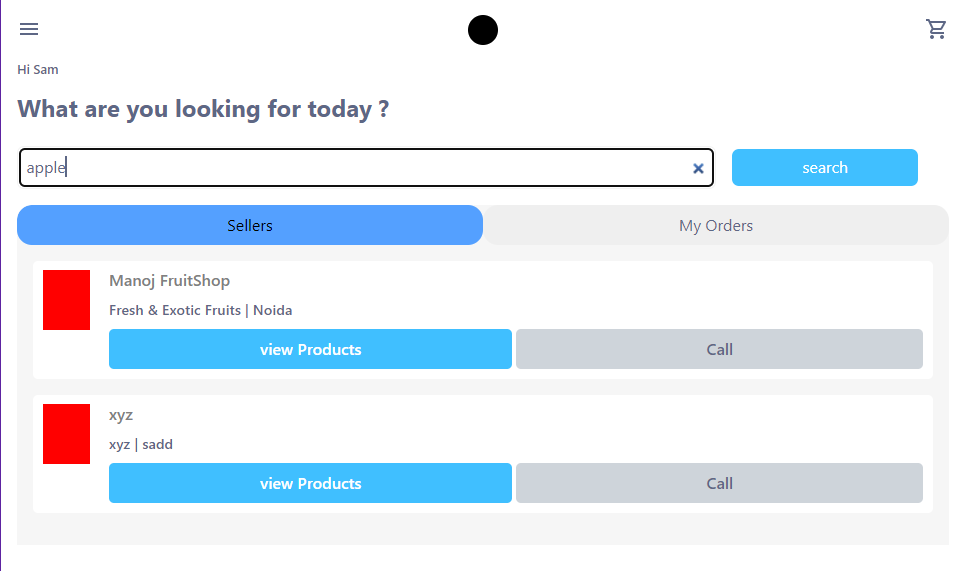


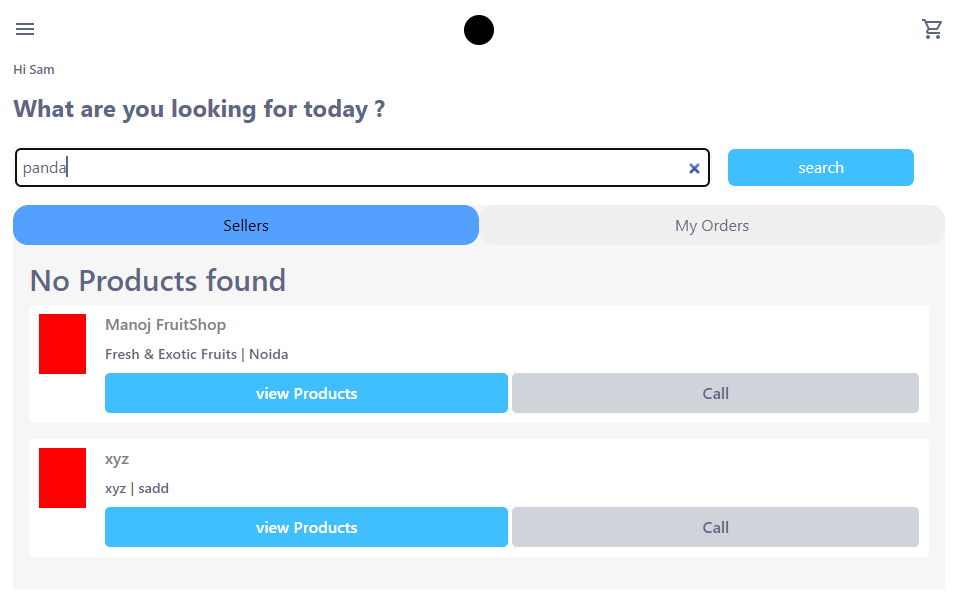
 





**Signup is valid if:**

1. Email is valid.
2. Phone Is valid phone.
3. Password and confirm Password fields match.
4. Email used is not used by any other customer to signup.
5. Phone is not used by any other customer to signup.
6. Fields are not empty.

**Signup is invalid if:**

1. Email is invalid.
2. Phone is invalid.
3. Password and confirm password fields do not match.
4. Email already exists in the DB.
5. Phone already exists in the DB.
6. Any of the fields are empty.

**Boundary Value Test Cases**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TC | Full Name | Email | Phone | Shop Name | Shop Address | Password | Confirm Password | Role | Expected Output | Actual Output | Status |
| 1 | Abc | [abc@test.com](mailto:abc@test.com) | 99999999999 | Abc shop | ABC Mall, Delhi | ABC@a#123 | ABC@a#123 | Seller | Seller Signedup | Seller Signedup | True |
| 2 | XYZ | [xyz@gmail.com](mailto:xyz@gmail.com) | 1234567890 | XYZ Shop | XYZ road, JJ colony | AAAA | AAAA | Seller | Seller Signedup | Seller Signedup | True |
| 3 | Reeha | [reeha@google.com](mailto:reeha@google.com) | 0000000000 | Home | Sector – 16, rohini | AAAA@1233 | AAAAA@1233 | Buyer | Buyer Signed up | Buyer Signed up | True |
| 4 | Saba | [saba@ceo.in](mailto:saba@ceo.in) | 1212121212 | Home | House no – 122, … | saba@123 | saba@123 | Buyer | Buyer Signed up | Buyer Signed up | True |
| 5 | Queen | [queen@elizabeth.com](mailto:queen@elizabeth.com) | 1111111111 | House | Villa no. 12,  Delhi… | AAAA | AAAA | Deliverer | Deliverer Signedup | Deliverer Signedup | True |
| 6 | ABC sharma | [Abc\_s@yahoo.com](mailto:Abc_s@yahoo.com) | 1000000000 | ABC bakers | Ambience Mall, … | ABC@a#123 | ABC@a#123 | Seller | Seller Signedup | Seller Signedup | True |
| 7 | KBC | [kbc@rediff.com](mailto:kbc@rediff.com) | 9870455572 | Home | Golden colony, … | kbccccccccc@ | kbccccccccc@ | Buyer | Buyer Signed up | Buyer Signed up | True |
| 8 | Amitabh Bachan | [Amitabh@cases.com](mailto:Amitabh@cases.com) | 9958937319 | Home | Appt – 19, delhi… | AAAA | AAAA | Buyer | Buyer Signed up | Buyer Signed up | True |
| 9 | My Name | [khan@outlook.com](mailto:khan@outlook.com) | 1231231231 | House | House – 10, 3rd floor, Delhi…. | 123456789 | 123456789 | Deliverer | Deliverer Signedup | Deliverer Signedup | True |

**\*\*\*Phone no. should be valid as OTP is sent for the verification purpose**

**\*\*\*Phone no. should be valid as OTP is sent for the verification purpose**

**As OTP is sent to verify the same**

**Robust Test Cases**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TC | Full Name | Email | Phone | Shop Name | Shop Address | Password | Confirm Password | Expected Output | Actual Output | Status |
| 1. | xyz | abc | 9870455572 | Abc\_shop | X | AAAA | AAAA | Inavlid | Inavlid | True |
| 2. |  |  |  |  |  |  |  | Invalid | Invalid | True |
| 3. | xyz | abc@gmail.com | 9870455572 | a | X | AAAA | AA | Invalid | Invalid | True |
| 4. | xyz | abc@gmail.com | 9870455 | abc | X | AAAA | AA | Invalid | Invalid | True |
| 5. | Abc | abc@gmail.com | 99999999999 | Abc shop | ABC Mall, Delhi | ABC@a#123 | ABC@a#123 | Valid | Valid | True |
| 6. | Xyz | abc@gmail.com | 1111111111 | Xyz shop | Xyz road | AA@123 | AA@123 | Invalid | Invalid | True |
| 7. | Abc | Abc1@gmail.com | 99999999999 | Abc shop | ABC Mall, Delhi | ABC@a#123 | ABC@a#123 | Invalid | Invalid | True |
| 8. | Abc | Abc1@gmail.com | 98704555572 | Abc shop | ABC Mall, Delhi | ABC@a#123 | ABC@a#123 | Valid | Valid | True |
| 9. | XYZ | abc@ | 00000000000 | xyz | xyz | AAAA | AAAA | Invalid | Invalid | True |
| 10. | XYZ | abc@test.com | 00000000000 | xyz | xyz | AAAA | AAAA | Valid | Valid | True |
| 11. | abc | Abc@test.com | XXXX | 123 | 123 | saba | saba | Invalid | Invalid | True |
| 12. | abc | Abc@test.com | 9313305723 | My shop | Karol bagh, shop no. 12 | abc@123@abc$ | abc@123@abc$ | Valid | Valid | True |
| 13. | reeha | reeha@google.com | 9899234567 | Pacific | Rohini, delhi | 123abc@omg | 123abc@omg | Valid | Valid | True |

**Conclusion:**

Test Cases for given code was made successfully.

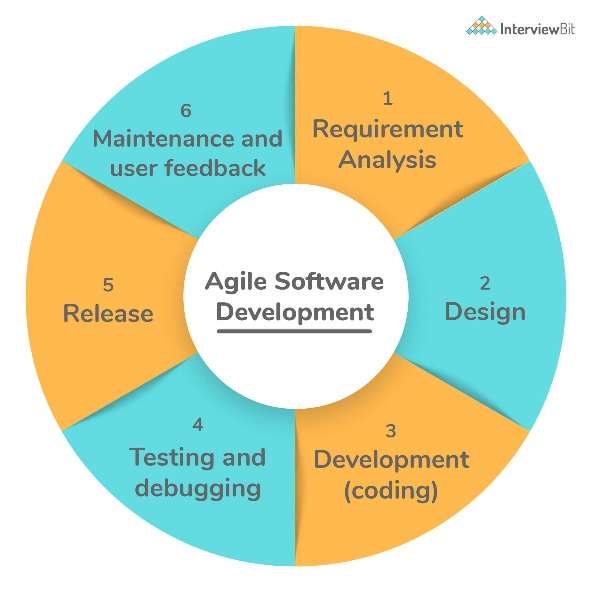
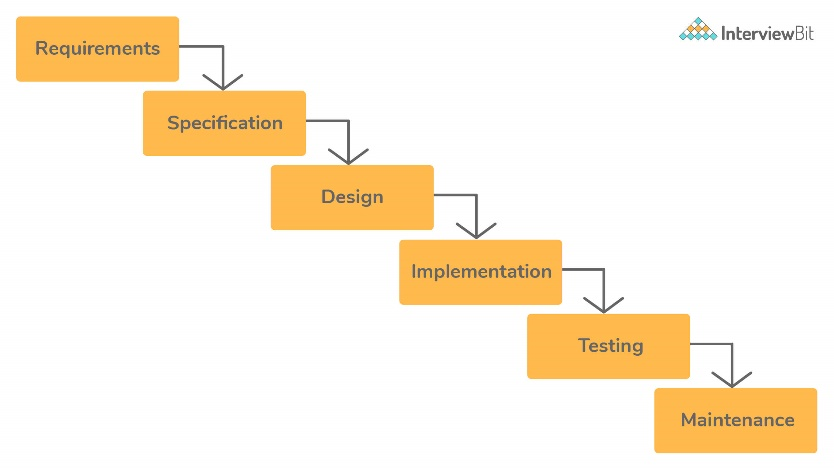
# **Viva Questions**

### **1. Explain the role of testing in software development?**

### Ans.

Software testing comes into play at different times in different software development methodologies. There are two main methodologies in software development, namely Waterfall and Agile.

In a traditional waterfall software development model, requirements are gathered first. Then a specification document is created based on the document, which drives the design and development of the software. Finally, the testers conduct the testing at the end of the software development life cycle once the complete software system is built.



Waterfall Software Development Model

An agile software development model works in small iterations. You test the software in parallel as it is getting built. The developers build a small functionality according to the requirements. The testers test it and get customer feedback, which drives future development.

### **2. How much testing is sufficient? Or, is it possible to do exhaustive testing of the software?**

### Ans.

It is impossible to exhaustively test software or prove the absence of errors, no matter how specific your test strategy is.

An extensive test that finds hundreds of errors doesn’t imply that it has discovered them all. There could be many more errors that the test might have missed. The absence of errors doesn’t mean there are no errors, and the software is perfect. It could easily mean ineffective or incomplete tests. To prove that a program works, you’d have to test all possible inputs and their combinations.

Consider a simple program that takes a string as an input that is ten characters long. To test it with each possible input, you’d have to enter 2610 names, which is impossible. Since exhaustive testing is not practical, your best strategy as a tester is to pick the test cases that are most likely to find errors. Testing is sufficient when you have enough confidence to release the software and assume it will work as expected.

### **3. Why developers shouldn’t test the software they wrote?**

### Ans.

Developers make poor testers. Here are some reasons why:

* They try to test the code to make sure that it works, rather than testing all the ways in which it doesn't work.
* Since they wrote it themselves, developers tend to be very optimistic about the software and don't have the correct attitude needed for testing: to break software.
* Developers skip the more sophisticated tests that an experienced tester would perform to break the software. They follow the happy path to execute the code from start to finish with proper inputs, often not enough to get the confidence to ship software in production.

However, it doesn't mean that developers shouldn't test the software before sending it to the tester. Developer testing helps find many bugs that are caused by programming errors. These are hard to find for a tester because they don't always have access to the source code.

### **4. What is functional testing?**

Ans.

Functional testing is a form of black-box testing. As the name suggests, it focuses on the software's functional requirements rather than its internal implementation. A functional requirement refers to required behavior in the system, in terms of its input and output.

It validates the software against the functional requirements or the specification, ignoring the non-functional attributes such as performance, usability, and reliability.

Functional testing aims to answer the following questions, in particular:

* Does the software fulfill its functional requirements?
* Does it solve its intended users' problems?

### **5. What is non-functional testing?**

Ans.

Non-functional testing tests the system's non-functional requirements, which refer to an attribute or quality of the system explicitly requested by the client. These include performance, security, scalability, and usability.

Non-functional testing comes after functional testing. It tests the general characteristics unrelated to the functional requirements of the software. Non-functional testing ensures that the software is secure, scalable, high-performance, and won't crash under heavy load.