# **TESTING:**

In this phase, we evaluate the functionalities of the system with the expected the specification and requirements. It ensures that the system includes all the features and function as per the requirements. Bugs were identified and removed in this process. This phase verifies whether the system was ready to use or not. This also check the functionality of the system.

Some of the technique to perform testing are:

* Black box testing – This type of testing is done to check the external mechanism i.e. functionality of the system. This is focused on the result i.e. output rather than internal mechanism. Validation is done to check the functionality of the system.
* White box testing – Internal mechanism of the system are checked in this type of testing. Verification is one of the tasks done in this testing.
* Unit testing – In this testing all the unit of the software are tested. Testing of whether all the individual units are working are not, are done in this phase. It helps to reuse the unit in another system. It will decrease the chance of system failure.
* Integration testing – Test are carried out in a group of units at once to produce an output.
* Regression testing – In this testing new components are added in the system. After that, system is checked whether the function and features are running or not.

To check the correctness of the built system, I performed two testing. Those are black box and unit testing.

Reason behind using black box testing are:

* Check the functionality of the system.
* Test are done from the view of end-users.
* Help to identify difference between requirements and final product.
* No need of technical knowledge i.e. programming.
* Test cases can be generated before development and after specification.

Problems of using black box testing:

* Have small test area i.e. test only small number of inputs.
* If specifications are not clear then designing test case are difficult.

Benefits of unit testing:

* Reuse of the codes are possible
* Easy to change and maintain codes.
* Better for rapid development
* Finding bugs are easier

Problems of using unit testing:

* Professional are required

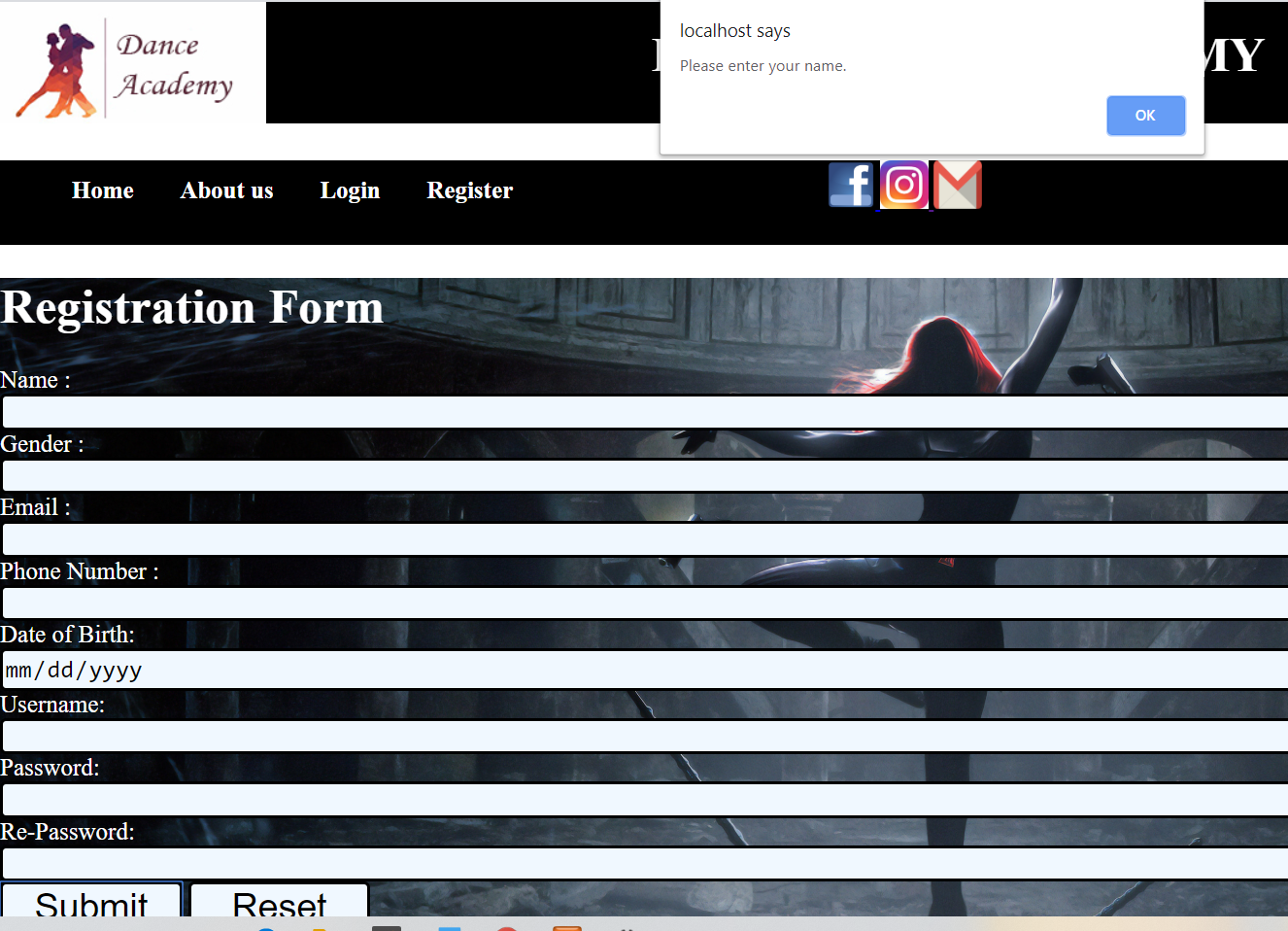


Figure : form validation

**Form validation**

|  |  |
| --- | --- |
| **Test ID** | 1 |
| **Test plan** | Check for validation in registration |
| **Expected Result** | Validation of form |
| **Actual Result** | Validate successfully |
| **Result** | Pass |

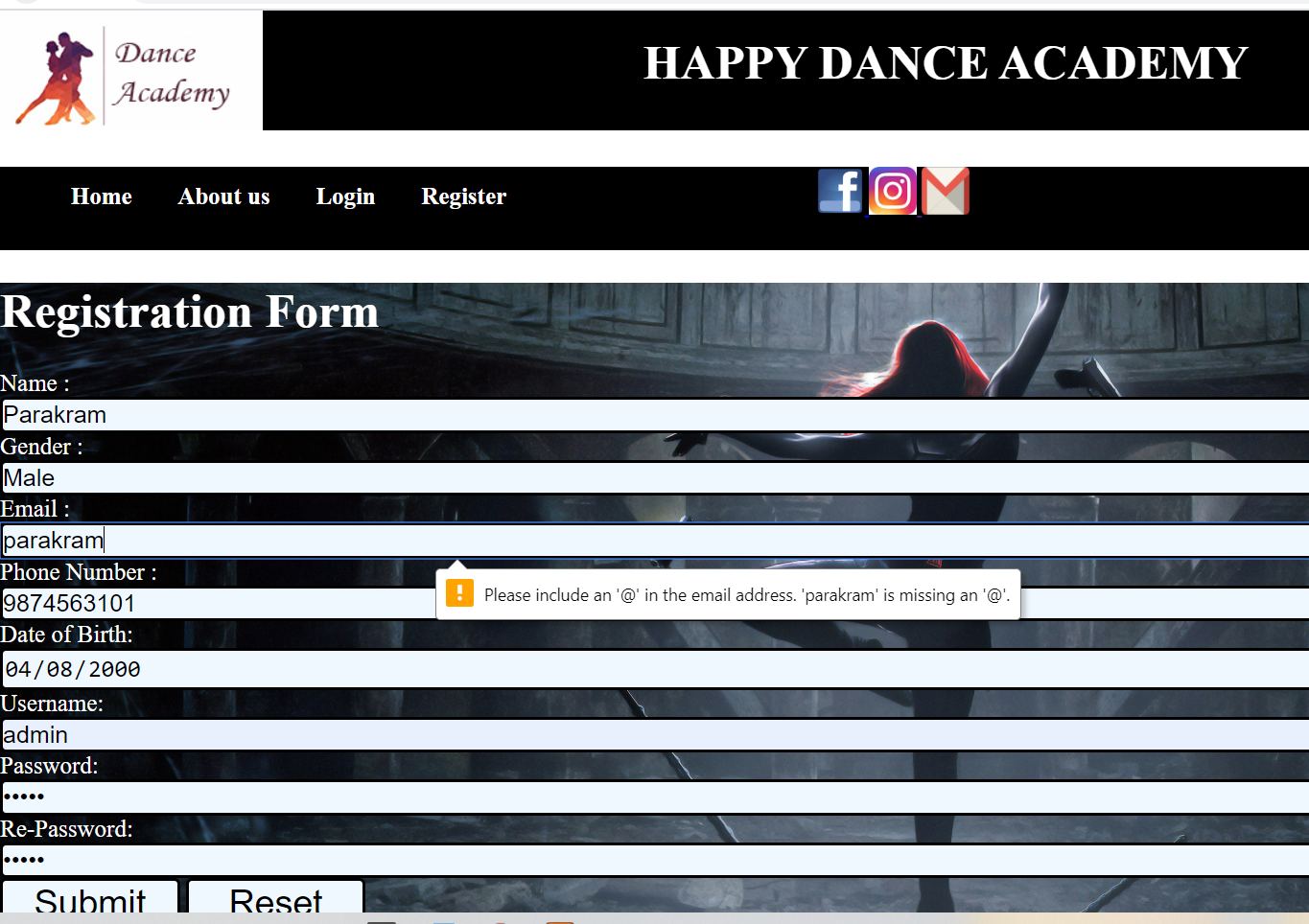


Figure :email validation

**Email validation**

|  |  |
| --- | --- |
| **Test ID** | 2 |
| **Test plan** | Check @ in email |
| **Expected Result** | Required @ in email |
| **Actual Result** | Validate successfully |
| **Result** | Pass |

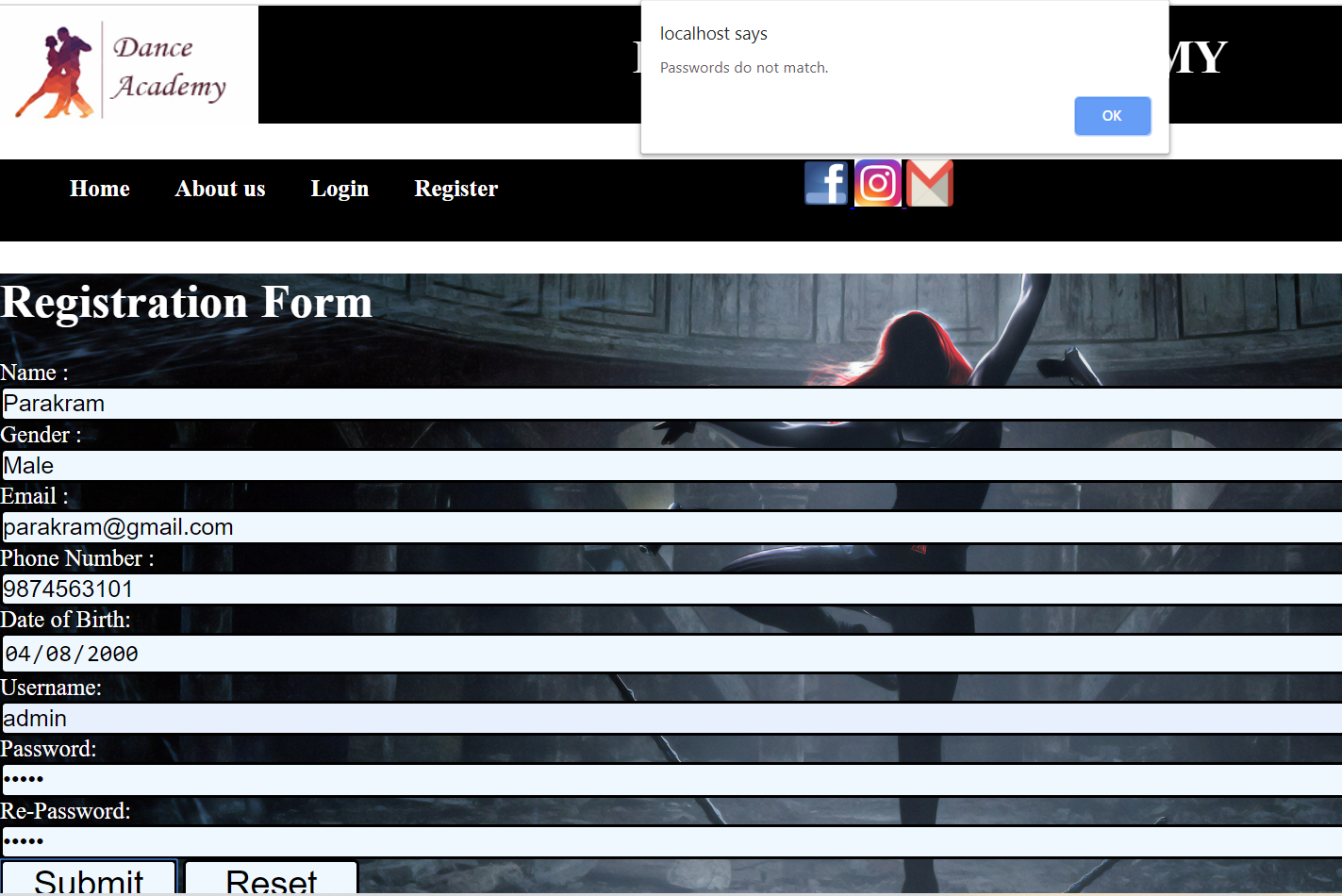


Figure :password match validation

**Password match validation**

|  |  |
| --- | --- |
| **Test ID** | 3 |
| **Test plan** | Check password and re-password |
| **Expected Result** | Required same password |
| **Actual Result** | Validate successfully |
| **Result** | Pass |

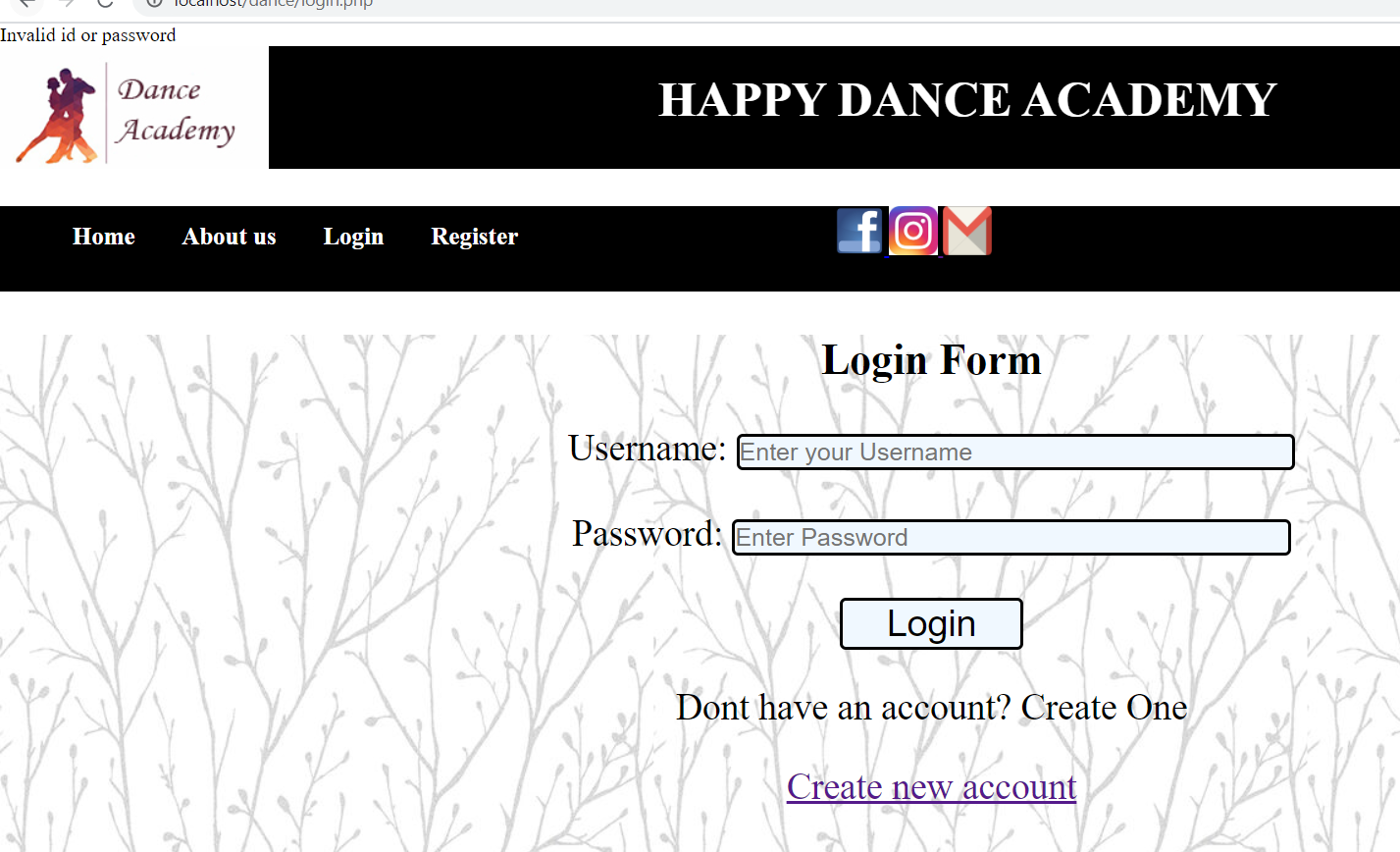


Figure :invalid username and password validation

**Username and password validation**

|  |  |
| --- | --- |
| **Test ID** | 4 |
| **Test plan** | Check username and password |
| **Expected Result** | Required valid username and password |
| **Actual Result** | Validate successfully |
| **Result** | Pass |

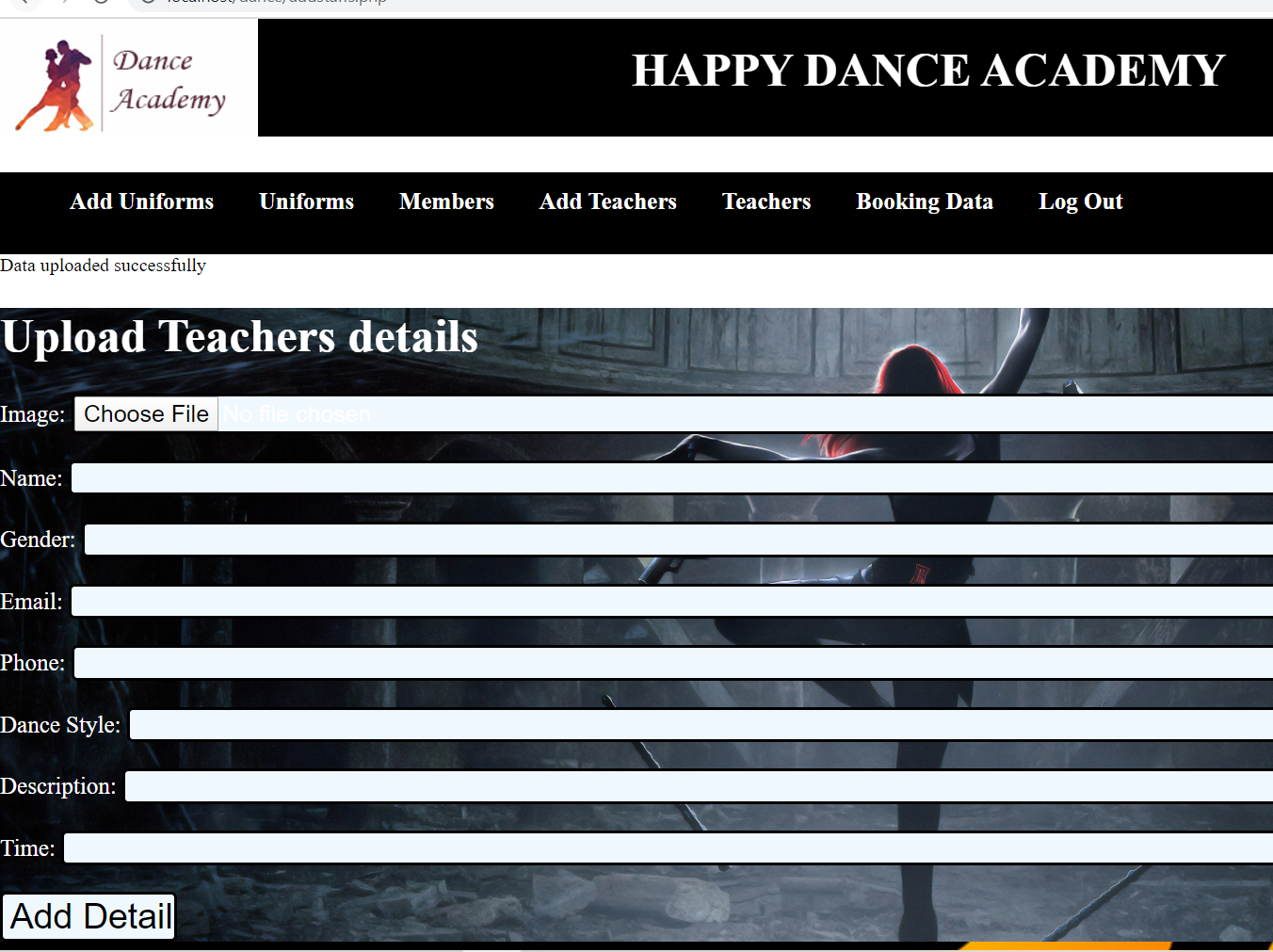


Figure :data upload

**Data store**

|  |  |
| --- | --- |
| **Test ID** | 5 |
| **Test plan** | Check data store in database |
| **Expected Result** | Data enter in database |
| **Actual Result** | Validate successfully |
| **Result** | Pass |