# Chapter-5 Testing

Testing is a process, which is used to evaluate the functionality of an application with an intent to find whether the developed software meets the specified requirements or not and also used to identify the problems or defects within the system.

## Unit Testing

Unittesting is a level of testing where individual units/ components of an application are tested. The purpose is to validate that each unit of the application performs as designed.

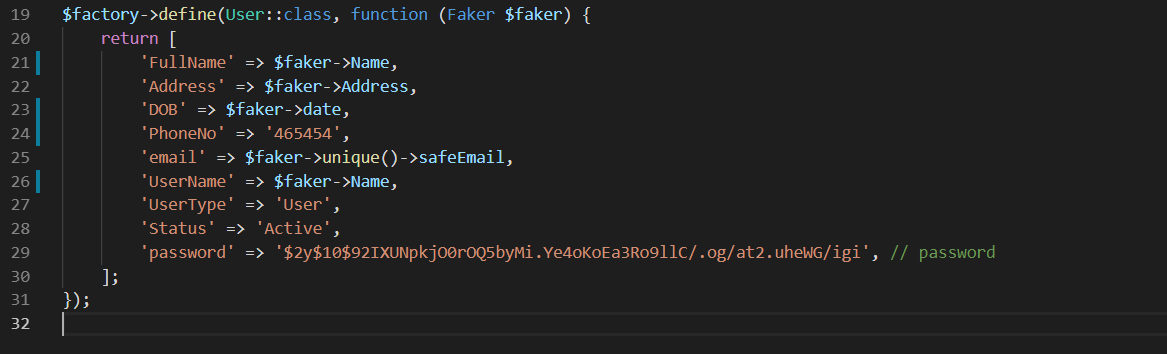


Figure 1 Fake Data for register



Figure 2 Testing of data

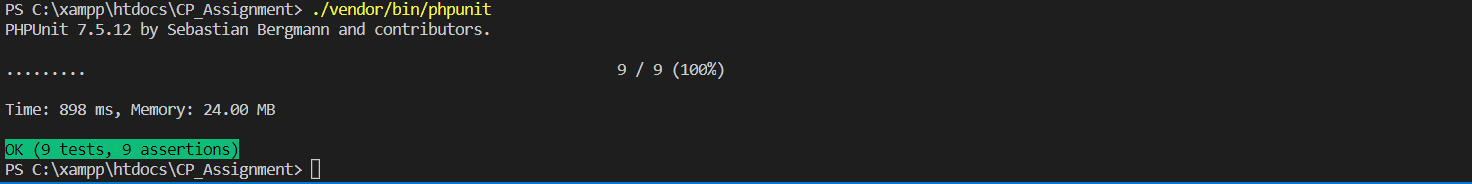


Figure 3 Test Passed

|  |  |
| --- | --- |
| Test Id | 1 |
| Test Class | User |
| Expected Outcome | User Register Successful |
| Actual Outcome | User Registration was successful |

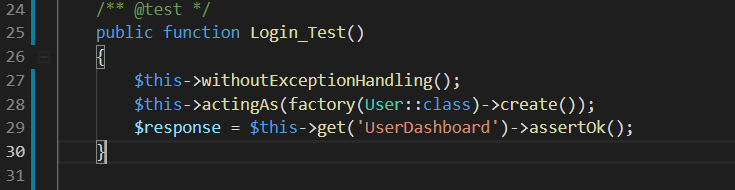


Figure 4 Login Test

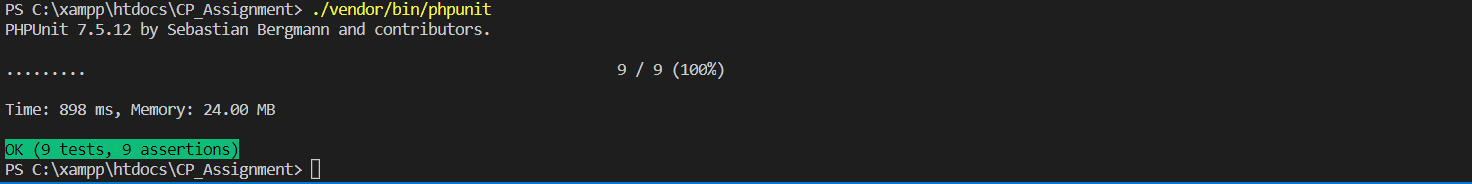


Figure 5 Test Pass

|  |  |
| --- | --- |
| Test Id | 2 |
| Test Class | User |
| Expected Outcome | Login Successful |
| Actual Outcome | Login was successful |

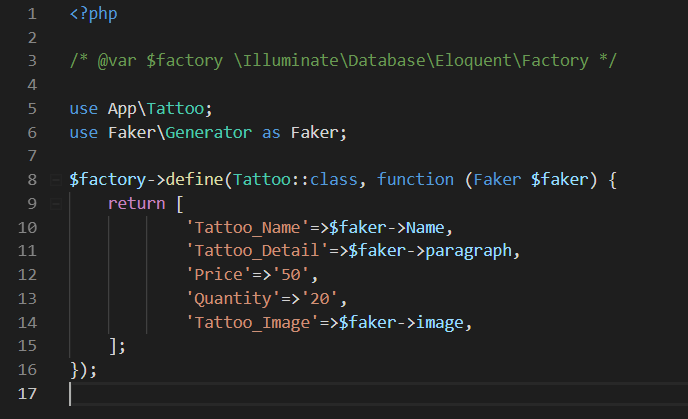


Figure 6 Fake Tattoo Data

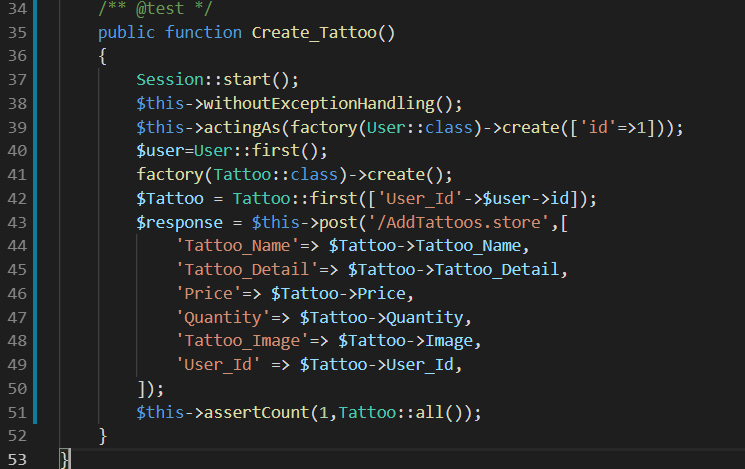


Figure 7 Create Tattoo Test

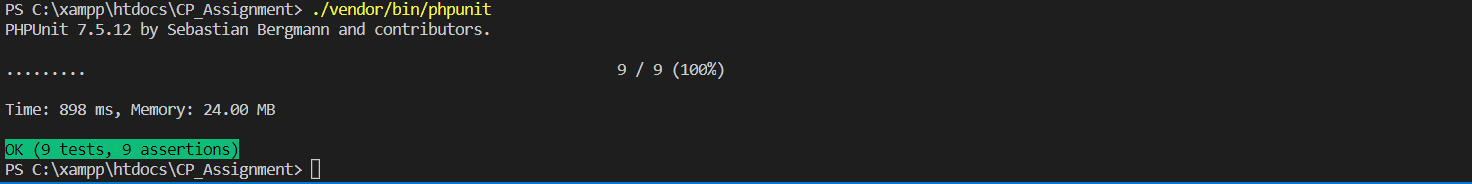


Figure 8 Test Pass

|  |  |
| --- | --- |
| Test Id | 3 |
| Test Class | Tattoo |
| Expected Outcome | Tattoo successfully created |
| Actual Outcome | Tattoo was successfully created |

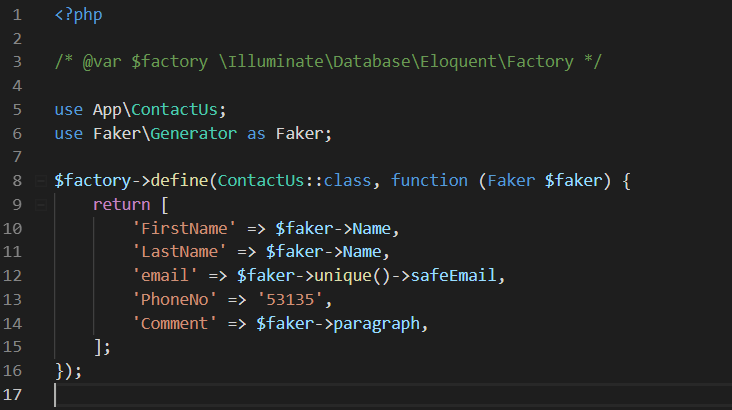


Figure 9 Fake Contact US Data



Figure 10 Create Contact US Test

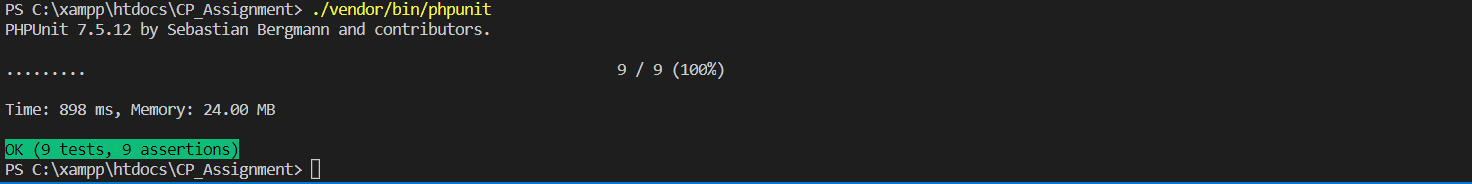


Figure 11 Test Pass

|  |  |
| --- | --- |
| Test Id | 4 |
| Test Class | Contact Us |
| Expected Outcome | Contact Us successfully created |
| Actual Outcome | Contact Us was successfully created |

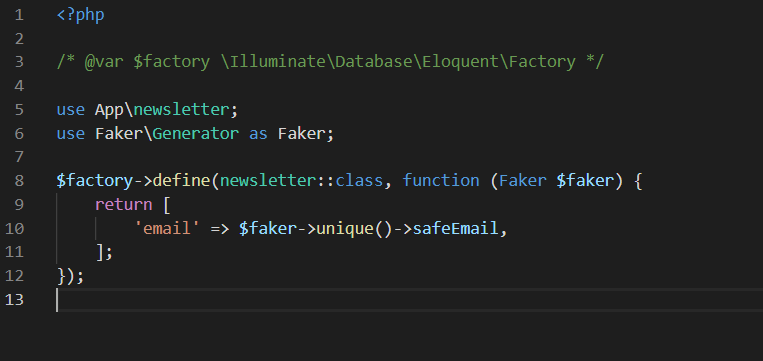


Figure 12 Fake Newsletter Data

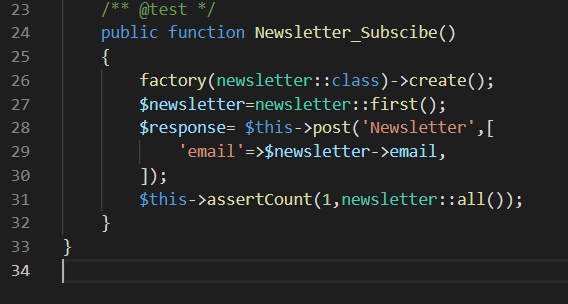


Figure 13 Newsletter Test

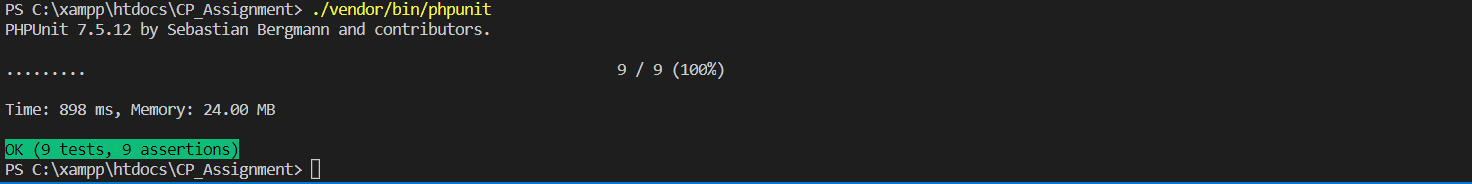


Figure 14 Test Pass

|  |  |
| --- | --- |
| Test Id | 5 |
| Test Class | newsletter |
| Expected Outcome | Newsletter successfully subscribed |
| Actual Outcome | Newsletter was successfully subscribed |

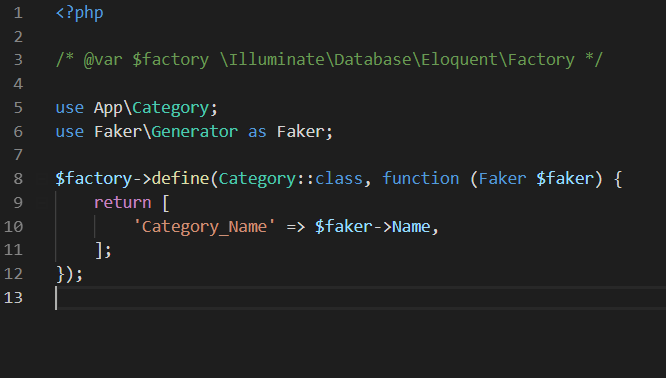


Figure 15 Fake Category Data

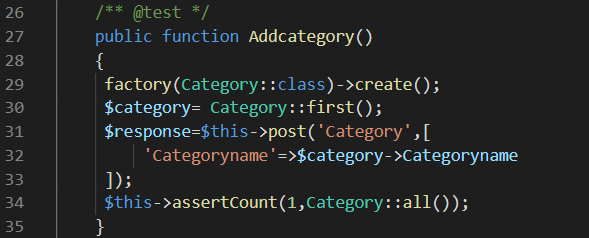
`

Figure 16 Add Category Test

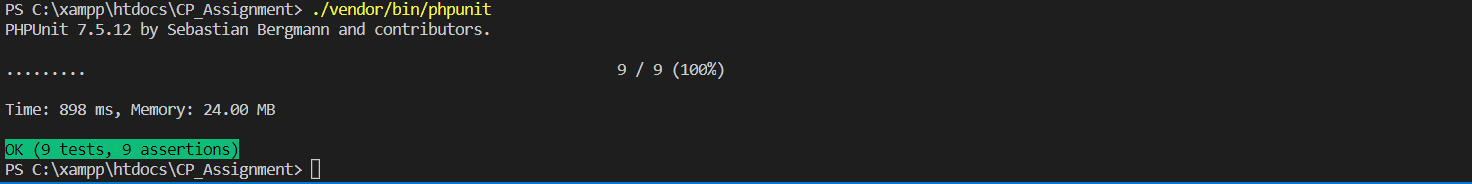


Figure 17 Test Pass

|  |  |
| --- | --- |
| Test Id | 6 |
| Test Class | Category |
| Expected Outcome | Category successfully created |
| Actual Outcome | Category was successfully created |



Figure 18 Buying Tattoo Test

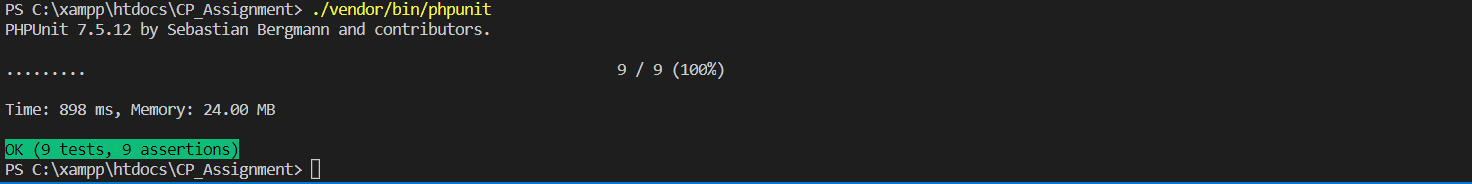


Figure 19 Test Pass

|  |  |
| --- | --- |
| Test Id | 7 |
| Test Class | buys |
| Test Data | Payment Method = “Esewa”, Price =”2000”, Quantity =”1”, Total =”2000”, Location =” asdasd”, Contact =” asdasd” |
| Expected Outcome | Tattoo successfully bought |
| Actual Outcome | Artist was successfully bought |

## Black Box Testing

Black-box testing is a method of application testing that examines the functionality of an application which does not concern about the internal codes or structures it mainly focuses on the user interface of an application. It concerns on whether the functionality works or not.

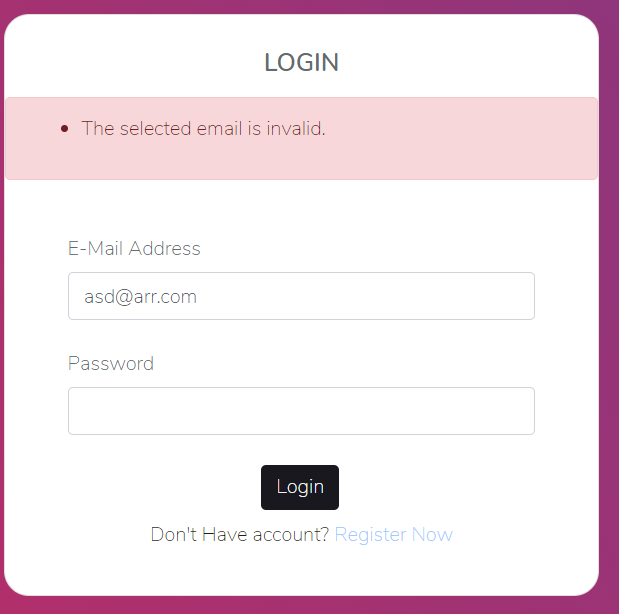


Figure 20 Login Fail with wrong email

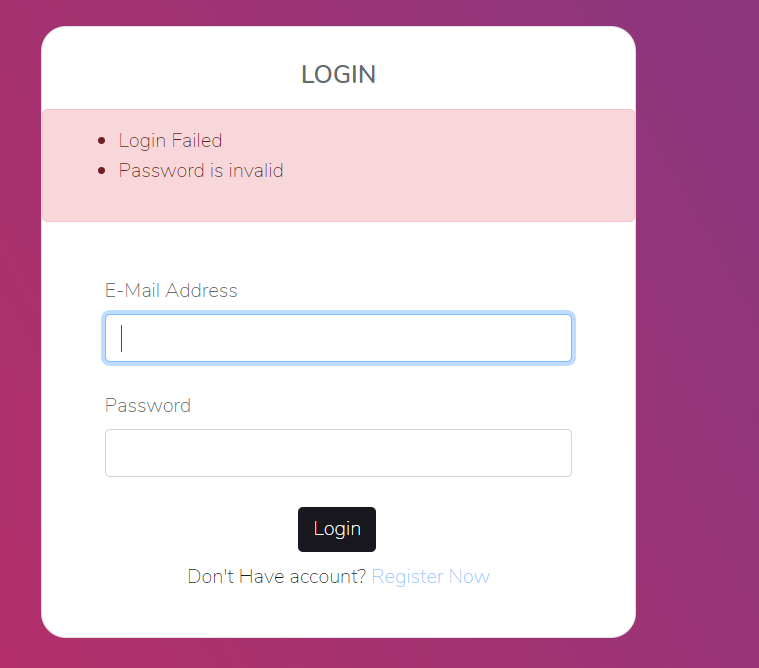


Figure 21 Login Failed with wrong password

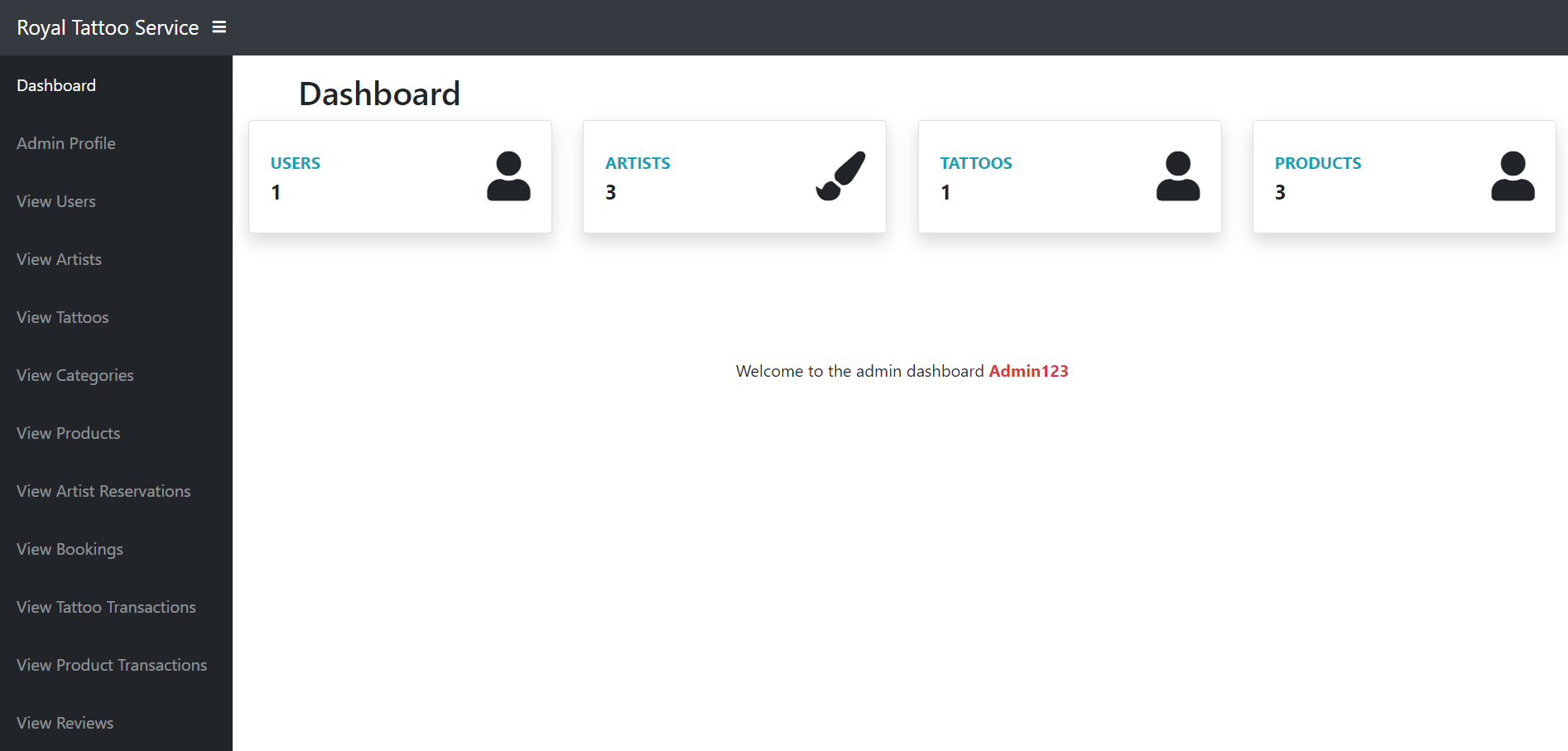


Figure 22 Dashboard After Login

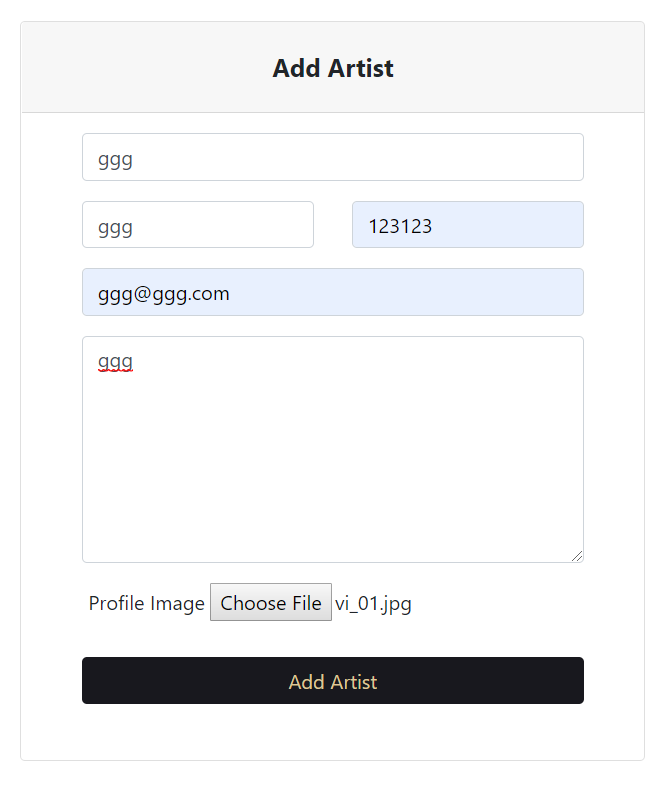


Figure 23 Add Artist Form

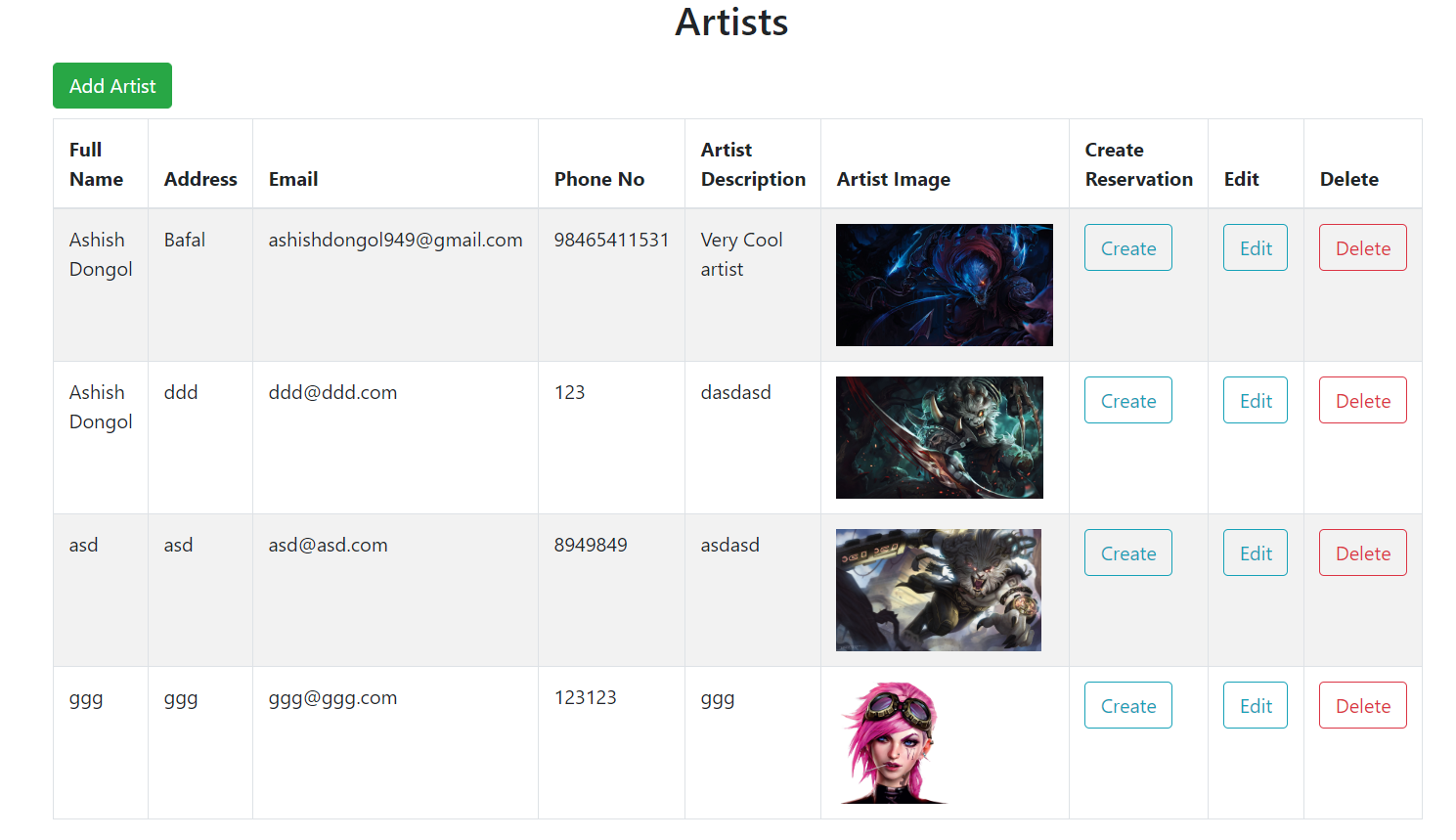


Figure 24 After Addition

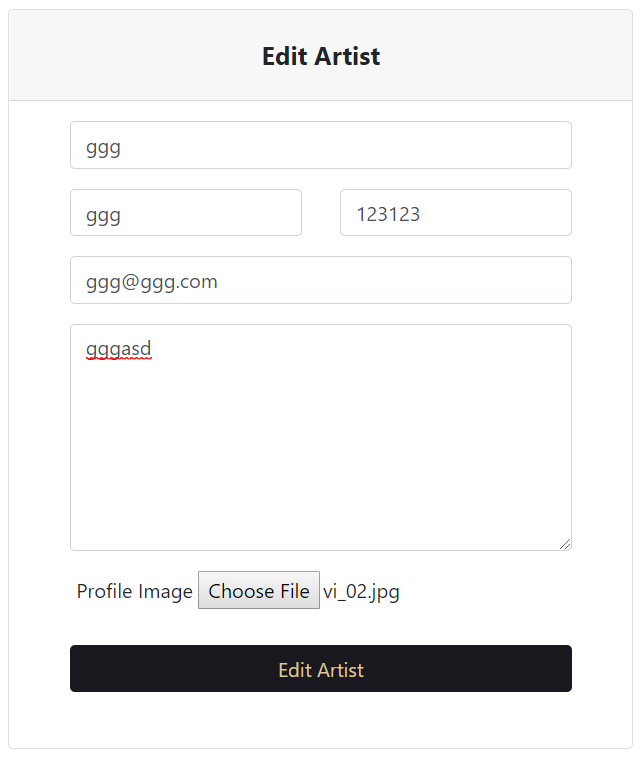


Figure 25 Edit Artist Form

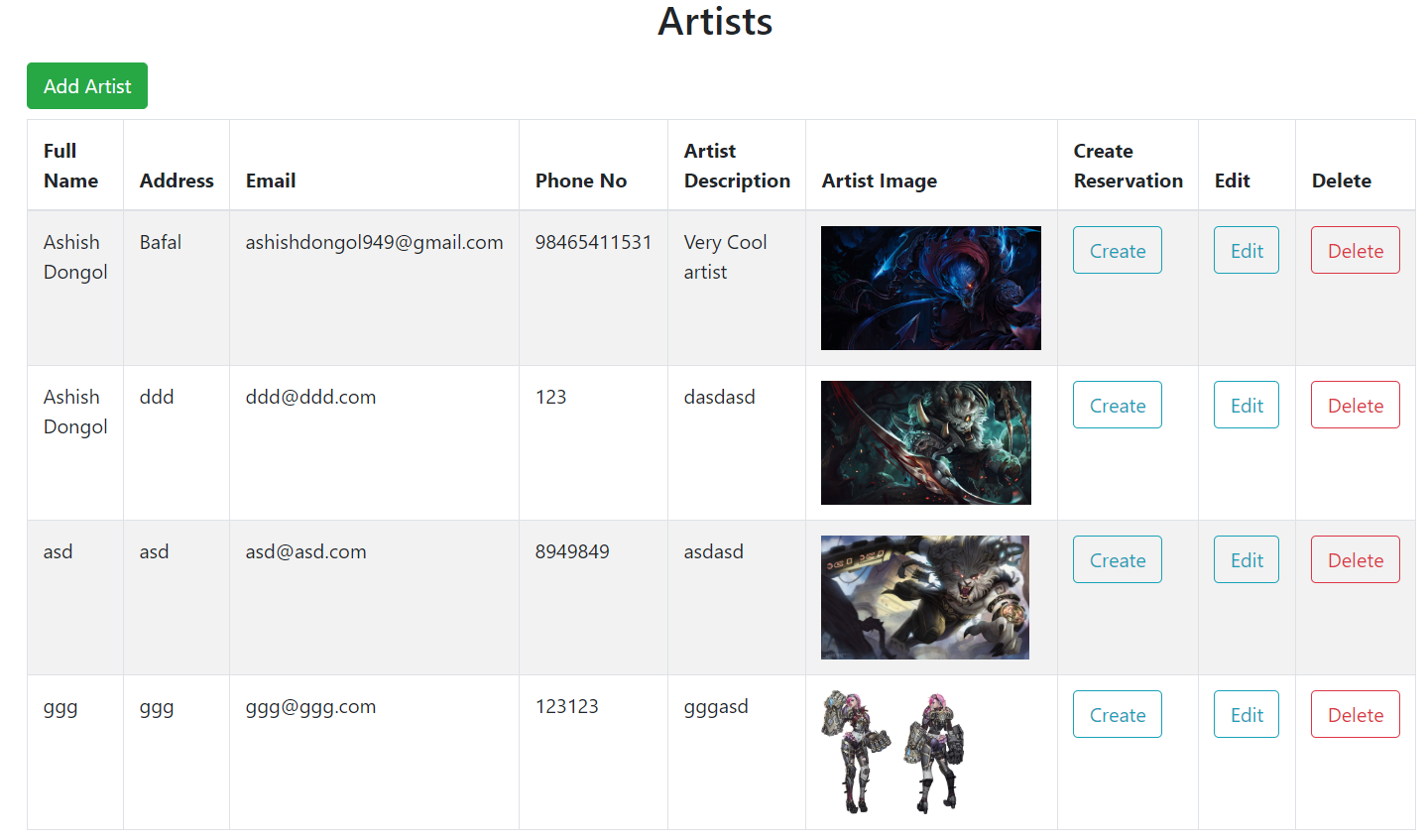


Figure 26 After Edit

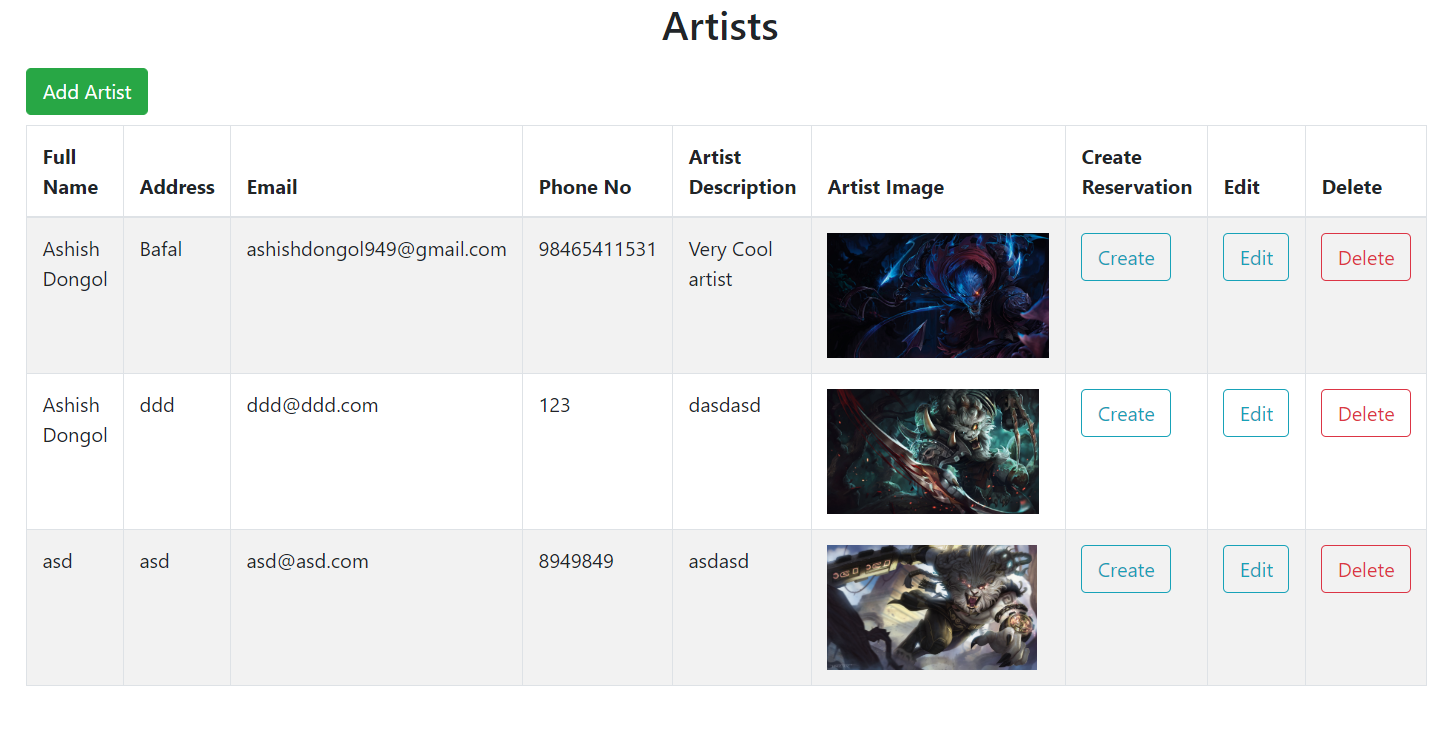


Figure 27 After Deletion

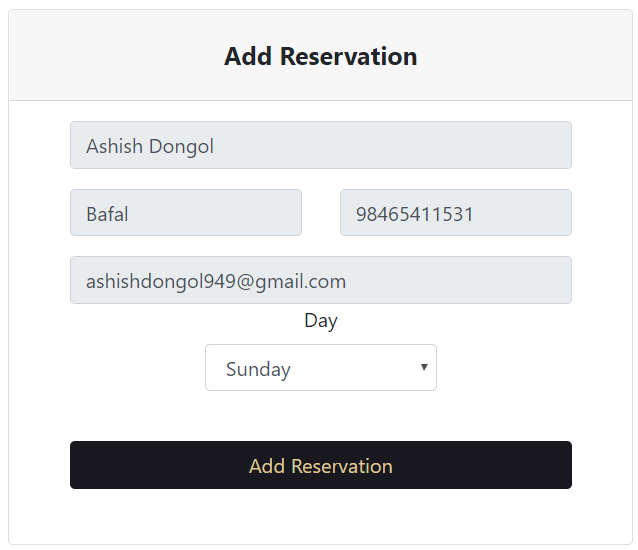


Figure 28 Reservation Addition Form

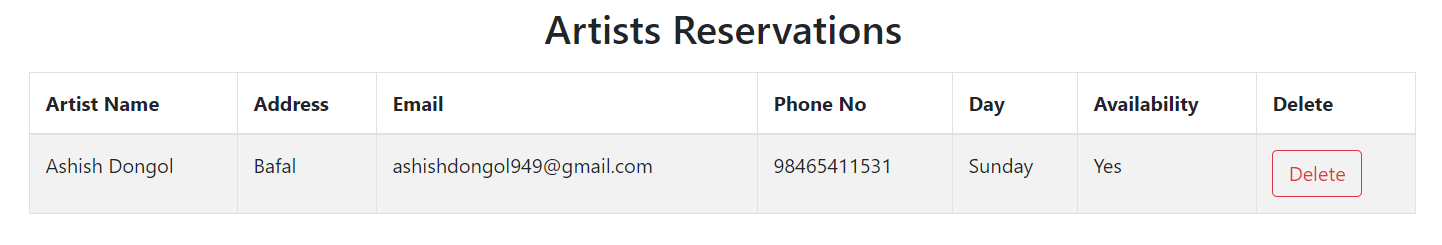


Figure 29 After Addition

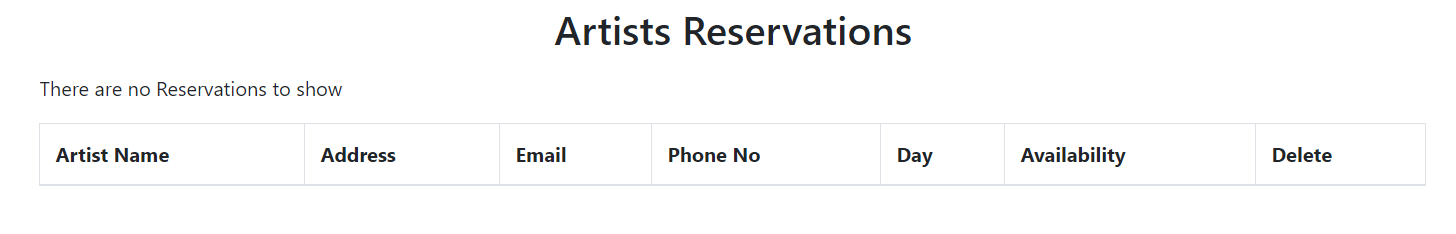


Figure After Deletion

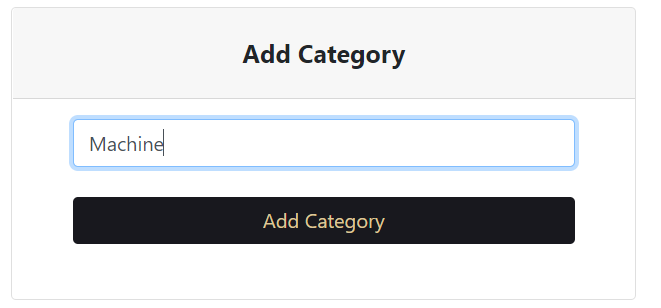


Figure 30 Add Category Form

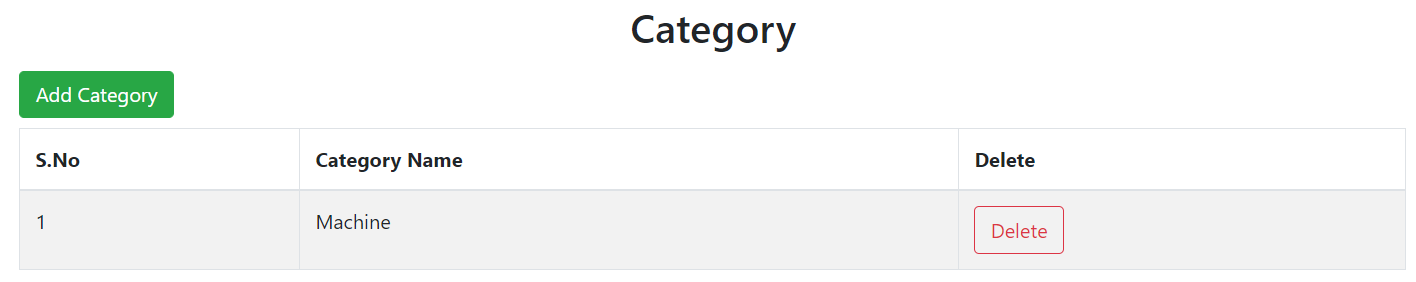


Figure 31 After Addition

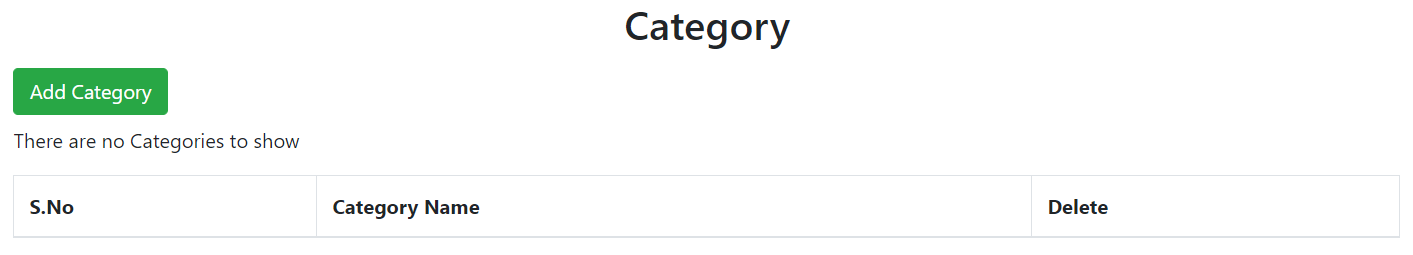


Figure 32 After Deletion

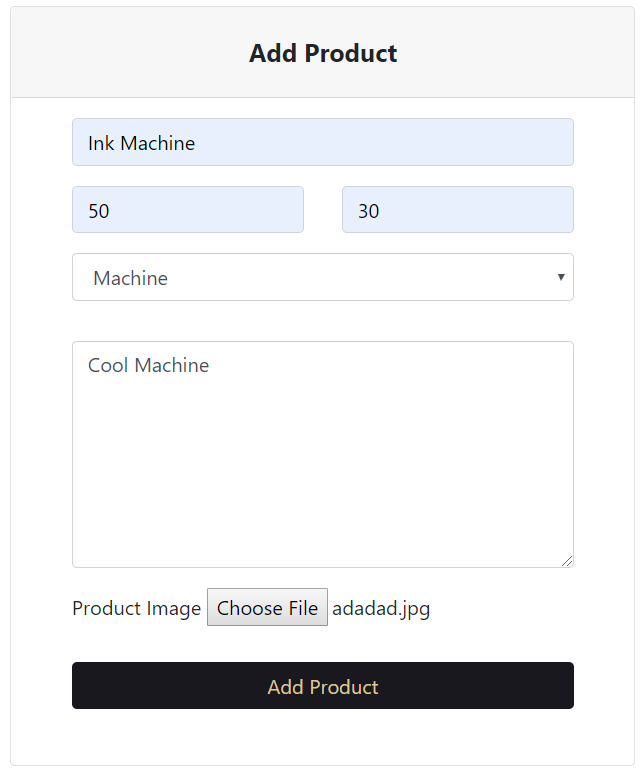


Figure 33 Add Product Form

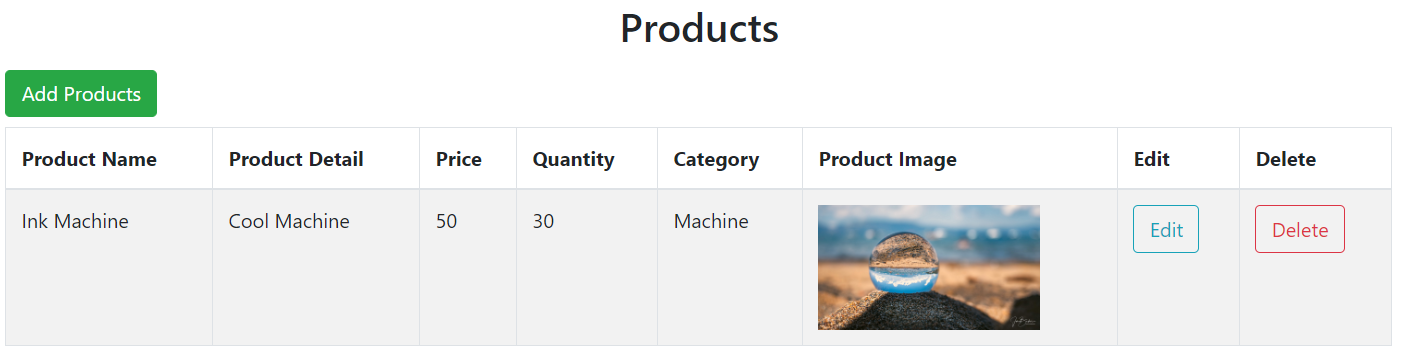


Figure 34 After Addition

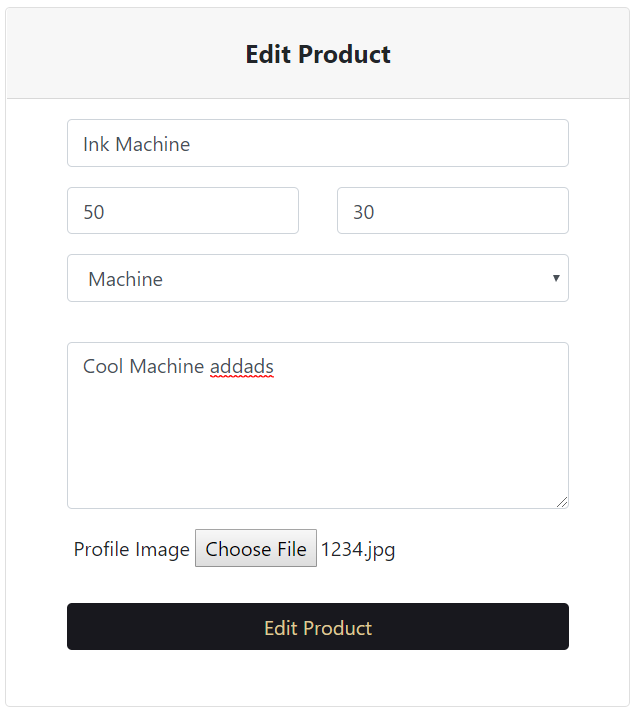


Figure 35 Product Edit Form

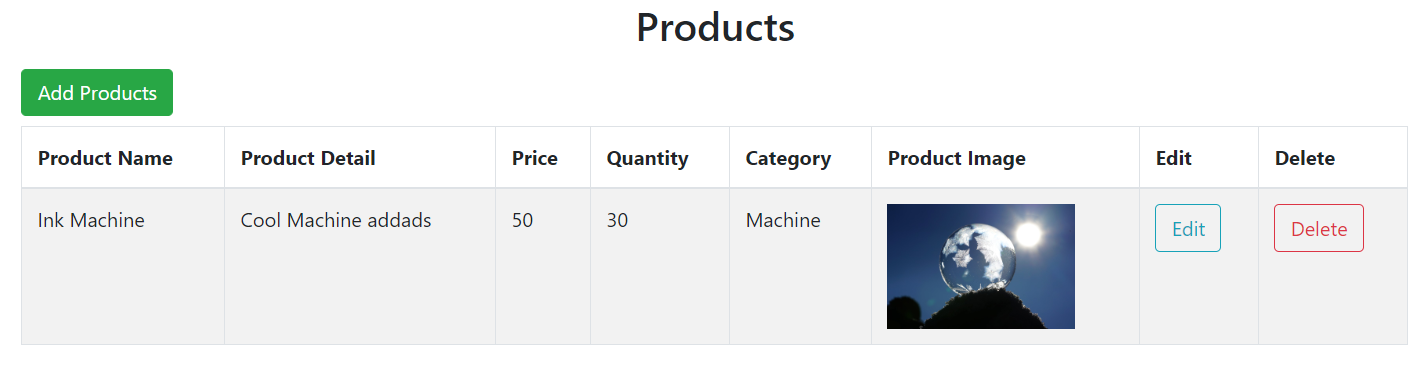


Figure 36 After Edit of Product

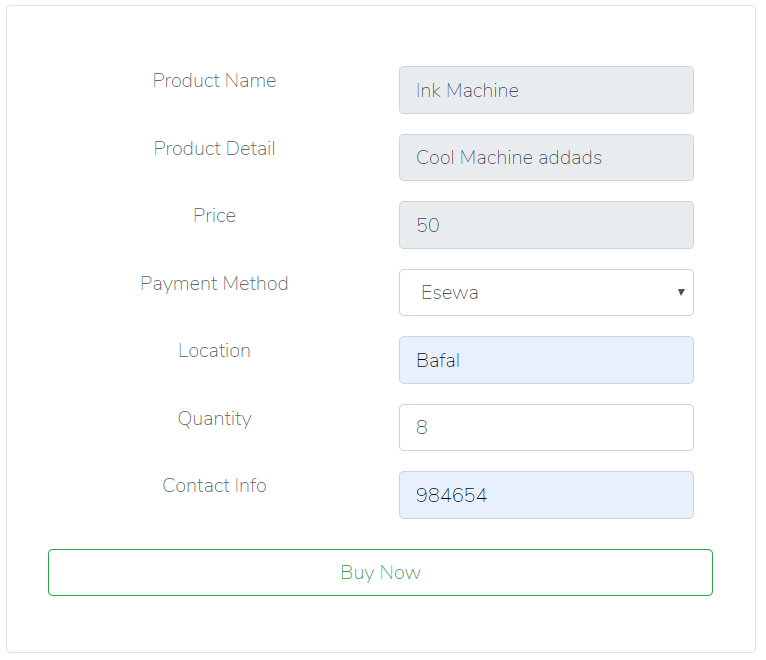


Figure 37 Product Buy Form

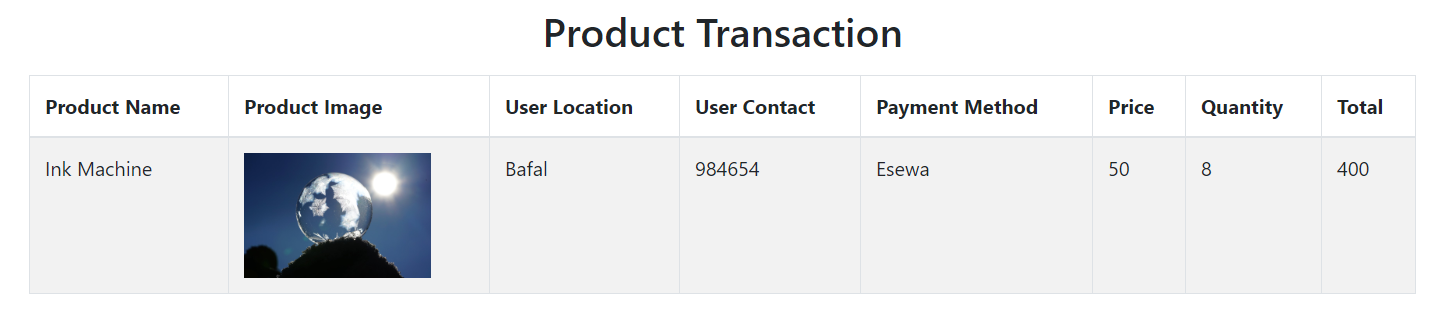


Figure 38 After Buying

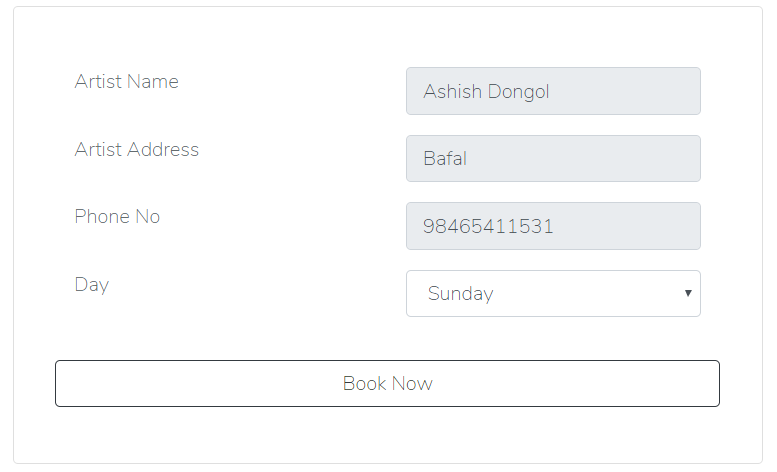


Figure 39 Artist Booking Form

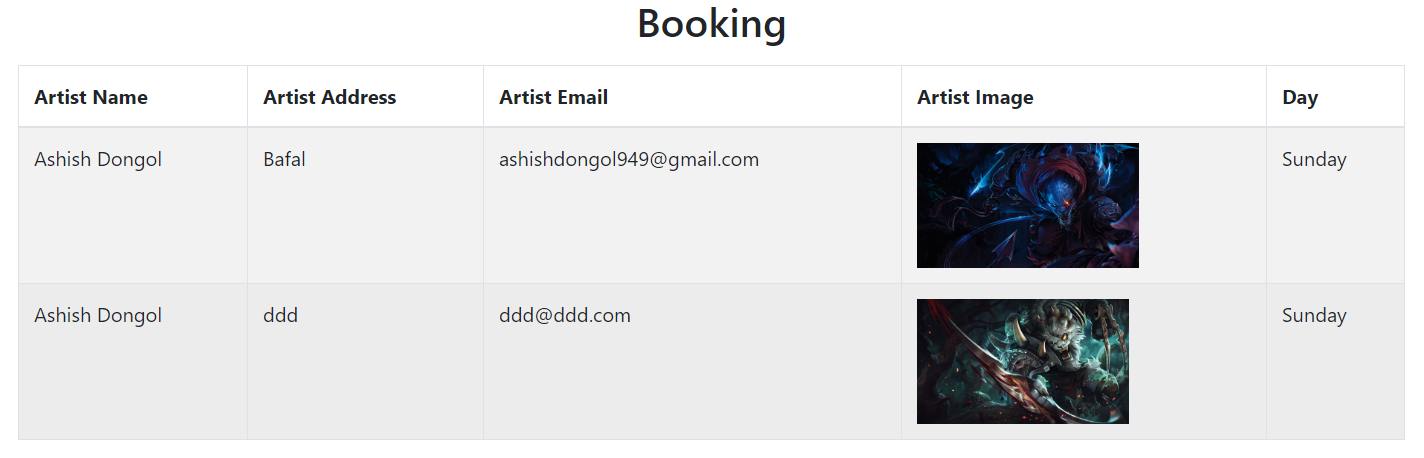


Figure 40 After Booking

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Description** | **Expected Result** | **Test Data** | **Actual Result** | **Status** | **Tester** | **Date/Time** | **Comments** |
| Admin Login | Login Successful | Email=admin@admin.com, Password=dasd | Login Fail | Fail | Ashish Dongol | 2019/06/30 | Wrong Password |
| Admin Login | Login Successful | Email=admin@admin.com, password=admin123 | Login Success | Pass | Ashish Dongol | 2019/06/30 |  |
| Artist Addition | Artist Successfully Added | Name=gg, address=ggg, phone=123123, [email=ggg@ggg.com](mailto:email=ggg@ggg.com), comment=ggg | Addition Successful | Pass | Ashish Dongol | 2019/06/30 |  |
| Artist Update | Artist Successfully Updated | Name=gg, address=ggg, phone=123123, [email=ggg@ggg.com](mailto:email=ggg@ggg.com), comment=ggasd | Update Successful | Pass | Ashish Dongol | 2019/06/30 |  |
| Artist Deletion | Artist Successfully Deleted | Id=4 | Deletion Successful | Pass | Ashish Dongol | 2019/06/30 |  |
| Reservation Addition | Reservation Successfully Added | Day = Sunday | Addition Successful | Pass | Ashish Dongol | 2019/06/30 |  |
| Reservation Deletion | Reservation Successfully Deleted | Id=1 | Deletion Successful | Pass | Ashish Dongol | 2019/06/30 |  |
| Category Add | Category Successful Added | Category name = Machine | Category Successfully Added | Pass | Ashish Dongol | 2019/06/30 |  |
| Category Delete | Category Successful Deleted | Id=1 | Category Delete Successful | Pass | Ashish Dongol | 2019/06/30 |  |
| Product Add | Product Successful Added | Ink Machine, 50, 30, Cool Machine | Addition Successful | Pass | Ashish Dongol | 2019/06/30 |  |
| Product Update | Product Successful Updated | Ink Machine, 50, 30, Cool Machine addasd | Update Successful | Pass | Ashish Dongol | 2019/06/30 |  |
| Buy Product | Product Successful Bought | Esewa, Bafal, 8, 984654 | Successfully bought | Pass | Ashish Dongol | 2019/06/30 |  |
| Book Artist | Artist Successful Booked | Day = Sunday | Successfully booked | Pass | AshishDongol | 2019/06/30 |  |