#### **Basis Path**

#### **Algorithm for User Registration:**

- -1- Start
- -2- Check if username and password are provided
  - -3- No -> Output "Username and password required"
  - -4- Yes -> Proceed
- -5- Check if username length < 5
  - -6- Yes -> Output "Username too short"
  - -7- No -> Proceed
- -8- Check if password length < 4
  - -9- Yes -> Output "Password too short"
  - -10- No -> Proceed
- -11- Register user
- -12- Redirect to "UserLogin.php"
- -13- End

# Flow Graph Representation:

```
[Start]
       I
    [Check Credentials]
    /
   No/
           \Yes
   /
            \
[Error: Credentials] [Check Username Length]
                   \
  I
            No/
                   \Yes
            /
                    \
            ٧
                      ٧
  [End] [Username Error] [Check Password Length]
            I
                     /
                    No/
                             \Yes
            ı
                    /
            ٧
                              ٧
                    ٧
                  [Password Error] [Register User]
           [End]
                         /
                                  \
                         /
                                   \
                      [Redirect] [End]
                       I
                       ٧
                      [End]
```

# Formula for Cyclomatic Complexity

#### Nodes (N):

- 1. Start
- 2. Check if username and password are provided
- 3. Error: "Username and password required"
- 4. Check if username length < 5
- 5. Username Error: "Username too short"
- 6. Check if password length < 4
- 7. Password Error: "Password too short"
- 8. Register user
- 9. Redirect to "UserLogin.php"
- 10. End

Total Nodes = 10

#### Edges (E):

- Start to Check credentials
- Check credentials to Error (No credentials)
- Check credentials to Check username length
- Check username length to Username Error
- Check username length to Check password length
- Check password length to Password Error
- Check password length to Register user
- Register user to Redirect
- Redirect to End
- Username Error to End
- Password Error to End
- Error credentials required to End

Total Edges = 12

Connected Components (P):

Since the program is a single connected module, P = 1.

Calculation:

Using the formula V(G) = E - N + 2P:

$$V(G) = 12 - 10 + 2 \times 1 = 4$$

## Interpretation

A cyclomatic complexity of 4 for the PHP user registration system is considered a good. This score suggests that the code is straightforward and manageable.

### 2. Equivalence Partition

This document outlines the Equivalence Partition and Boundary Value Analysis for the user

registration feature of a PHP project. This testing approach helps in identifying efficient and effective

test cases, reducing redundancy while ensuring comprehensive coverage of functionality.

#### **Equivalence Partitions**

Input Field	Partition	Valid?
Username	5-15 alphanumeric	Yes
Username	< 5 characters	No
Username	> 15 characters	No
Username	Special characters	No
Password	4-12 characters	Yes
Password	< 4 characters	No
Password	> 12 characters	No

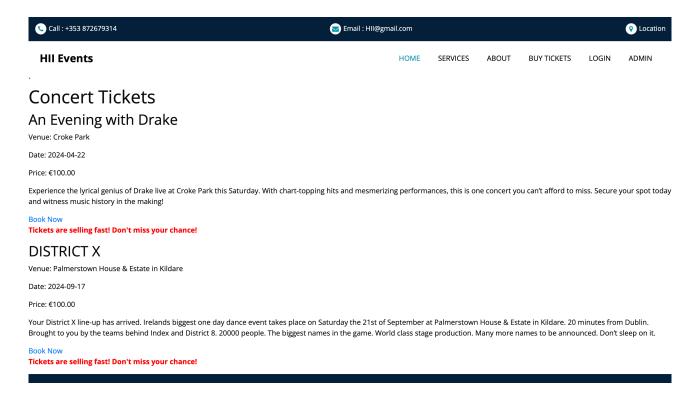
## **User Interface Testing**

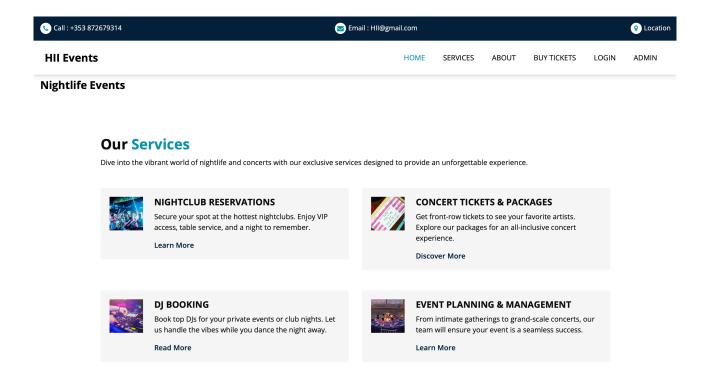
#### Consistency

Headers and Footers: Each page features the same layout, ensuring easy navigation and familiarity.

Colour Schemes: We use a consistent palette to enhance brand recognition and visual cohesion.

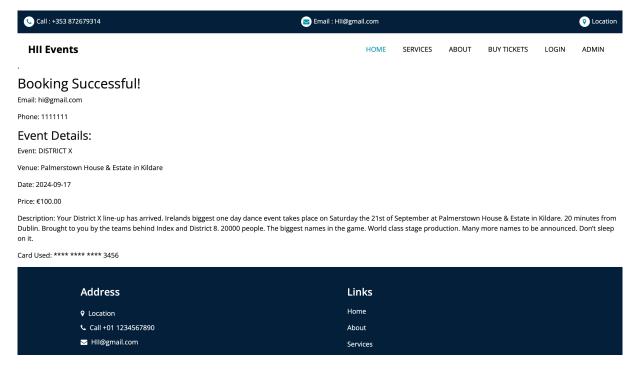
Textual Presentation: All text follows the same style guide, which improves readability and aesthetic appeal.





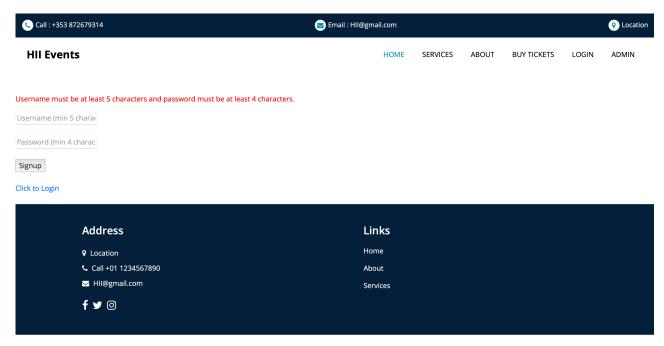
#### **Feedback**

Form Submission Confirmation: Users receive a clear confirmation message upon the successful submission of forms. This reassures them that their data has been processed.



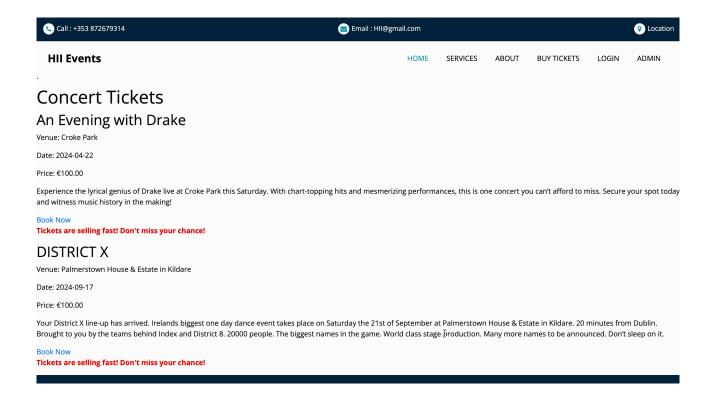
Error Messages: When users encounter errors, descriptive messages are displayed, guiding them on how to correct the issue.

These messages are designed to be helpful and constructive, reducing user frustration.



© All Rights Reserved By Free Html Templates

Interactive Tooltips and Hover Effects: Tooltips appear when users hover over interface elements that may require further explanation. This feature is particularly useful for enhancing understanding without cluttering the interface.



### **Validation Testing**

**Username Uniqueness**: The script checks if the username already exists in the database, ensuring that each user has a unique identifier. This is achieved by querying the database for the count of users with the specified username and confirming that it is zero.

**Password Strength**: It validates the strength of the password by ensuring it is at least 8 characters long and includes both letters and digits. This helps in protecting user accounts by enforcing the creation of secure passwords.

**Event Date Validation**: The script confirms that an event date is in the future, preventing the scheduling of events in the past and ensuring logical consistency in event management.

**Email Format Validation**: It verifies that the email provided follows a proper format, using PHP's built-in **filter\_var** function with the **FILTER\_VALIDATE\_EMAIL** filter. This step is crucial for maintaining valid and usable contact data.