

Malicious Payload Detector

**Object Oriented Software Development**

Daffodil International University

**Under The Guidance of:**

Name:

Designation: Lecturer

**Submitted By:**

Name:

ID:

Section:

Batch:

­­

**Course Code:**

*Abstraction:*

With the development of Web application, cross-site scripting attacks or malicious payload injection on input field have been rapidly increasing, and the technique of those attacks is constantly updating. There are some special advanced attacks such as the one based on encoding. This report presents a dynamically access control method to prevent them by means of the Cross Site Scripting attacks or malicious payload injection , making up the existing approach being lack of the practicality, and the experimental results verify the feasibility and practicality of our protection mechanism.

This project presents a cross-site scripting attacking detection method, which is based on improved white list. It can fulfill the rapid detection of cross-site scripting attacks by using   
some steps, for example, creating a white list signature, detecting suspicious strings,   
marking attack vectors and so on. Experiment shows that compared with the traditional   
defense method of XSS attack, the new defense method can reduce the false alarm rate of   
detection, and improve the time performance effectively.

*ACKNOWLEDGEMENT*

*First and foremost we would like to thank* Md. Mushfiqur Rahman *sir for giving us a chance to work in this project and giving us an opportunity to enhance our technical skills and performance. We would like to thank our parents and teachers for supporting us during our academic career, who are always support us so that we could focus on our project work. Finally, I would like to thank all of them whose names are not mentioned here but have helped us in some way to accomplish the work.*

*Table of Contents*

### ***Project summary***

### Aims …………………………………………………………………………………………… 4

### Object …………………………………………………………………………………………… 4

### Outcomes ………………………………………………………………………………………… 4

### Significance ……………………………………………………………………………………… 4

1. *Background* …………………………………………………………………………………………… 4-5
2. *Dataflow of this project* ……………………………………………………………………………. 5
3. *Screen-Shots*  
   4.1 User Interface ………………………………………………………………………………………… 5-6   
   4.2 Admin Panel ………………………………………………………………………………………… 7   
   4.3 Class Diagram ………………………………………………………………………………………… 8  
   4.4 ER Diagram …………………………………………………………………………………………… 8
4. *Coding Screen Shots*  
   5.1 Inheritance …………………………………………………………………………………………… 9-11  
   5.2 Encapsulation ……………………………………………………………………………………… 11-13
5. *Architecture ………………………………………………………………………..* 14-15

### 1. Project summary

#### **1.1 Aims:** Cybersecurity is the body of technologies, processes and practices designed to protect networks, computers, programs and data from attack, damage or unauthorized access. In a computing context, security includes both cybersecurity and physical security.

#### Ensuring cybersecurity requires coordinated efforts throughout an information system. Elements of cybersecurity include. 1. Application security 2.Information security that’s why at this time I’m working with web application security.

1.2 Object:  
 Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious payload into a web application. XSS is amongst the most rampant of web application vulnerabilities and occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.  
By using this project I will try to escaping web application from the severe of injecting malicious payload attack.

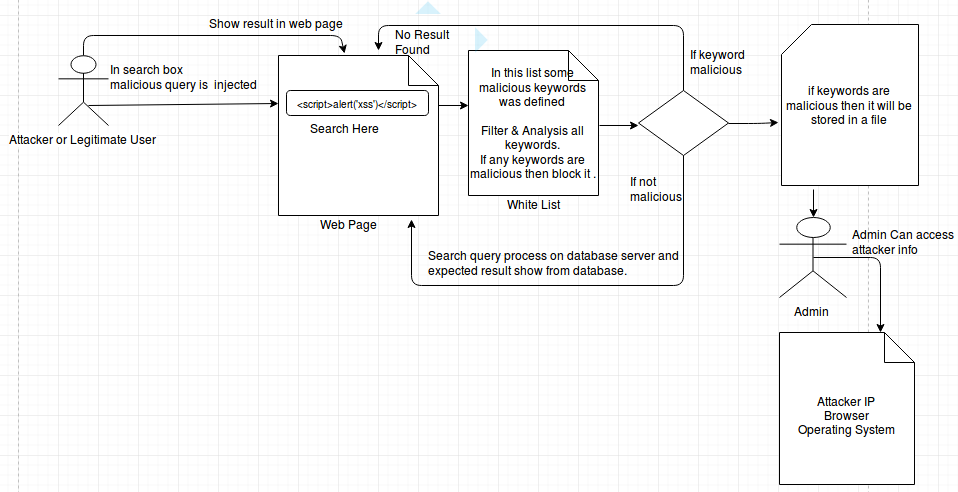
1.3 Outcomes:  
 By preventing website from injection malicious payload we also protect web application from other cyber threat like as,  
 > Prevent malicious payload or script execution  
 > Prevent redirecting to malicious server  
 > Preventing data manipulation  
 > Prevent session and cookie hijacking  
 > Prevent data theft

1.5 Significance:  
 Cyber threat Questioned about our existence and information security on cyber space. So it’s time to protect our web application from cyber threat as well as protect web application from injection malicious payload which will protect our online business, assets and user credentials. So this project protect cyber threat and user credentials by preventing malicious payload Injection.

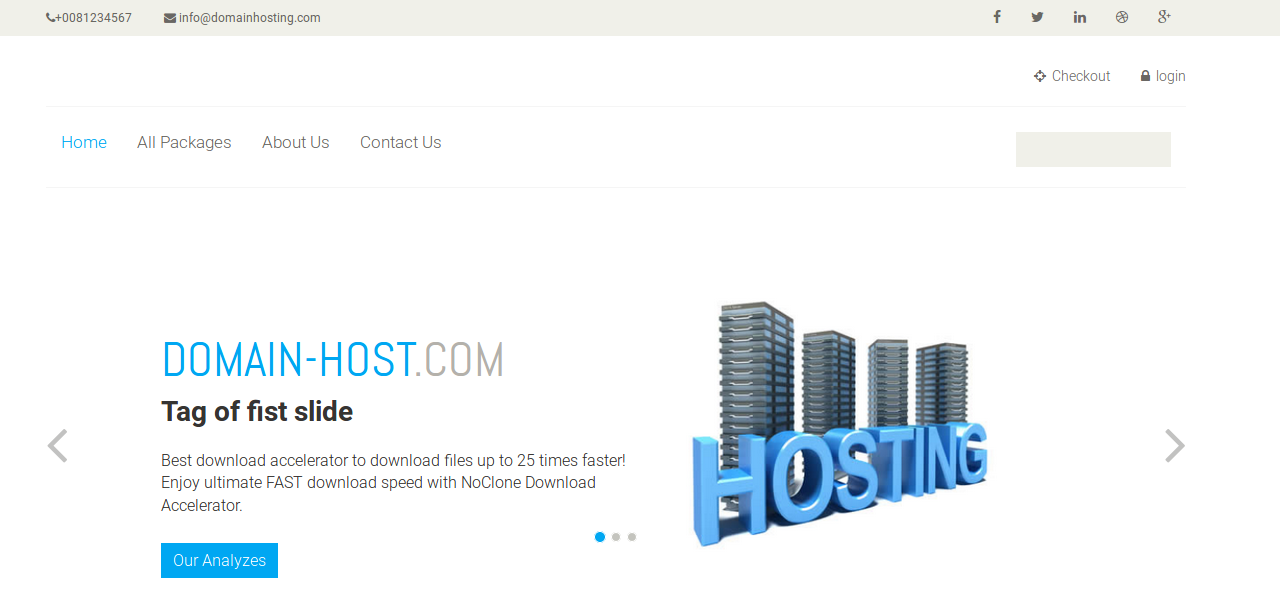
2.Background  
 XSS or malicious payload are very common and severe attacks imposed on web applications and web systems. Web applications vulnerable to these attacks allow attackers to execute script in the user’s browser. User interface has always been considered as the most vulnerable point of any web system. Both these attacks are implemented by violating the users trust on the applications and services they use on a day to day basis. Web applications include programs that reside in the web application server’s databases and CSRF attacks are implemented by manipulating the connection between user and server and tricking the user and server to run unauthorized scripts. These scripts collect the user's security sensitive data and pass them on to the attackers for further attacks. But still now there is no efficient method that can prevent XSS or malicious payload injection in web application input field and if some preventation method are available but there are some dependencies or limitations. That’s why I’m interested to working with this and contribute something that will performed better than before.

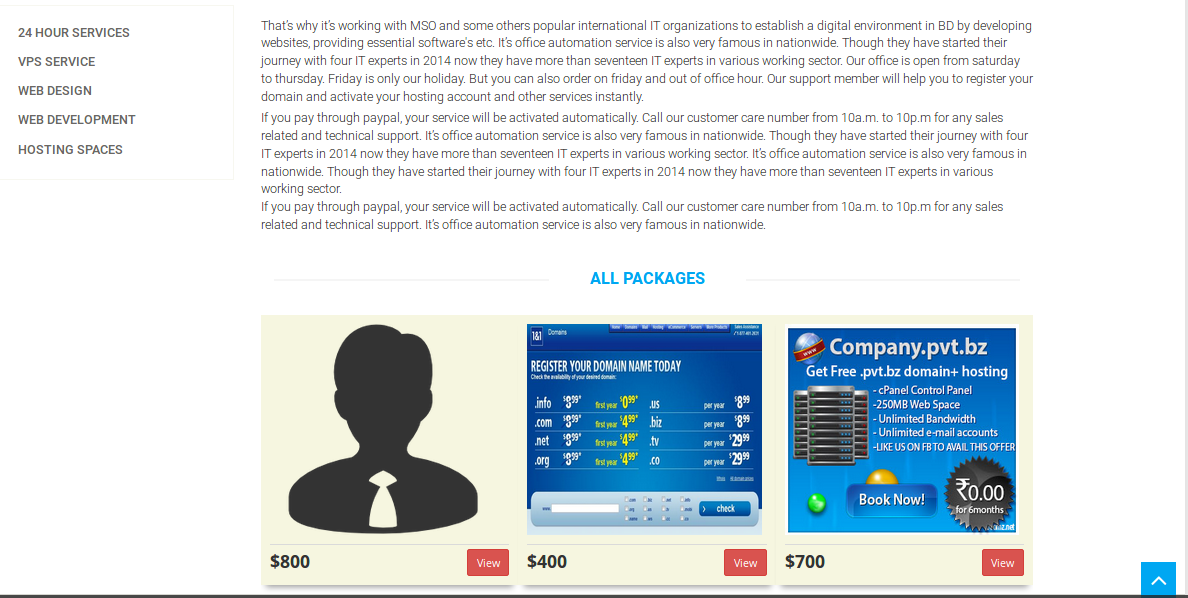
.

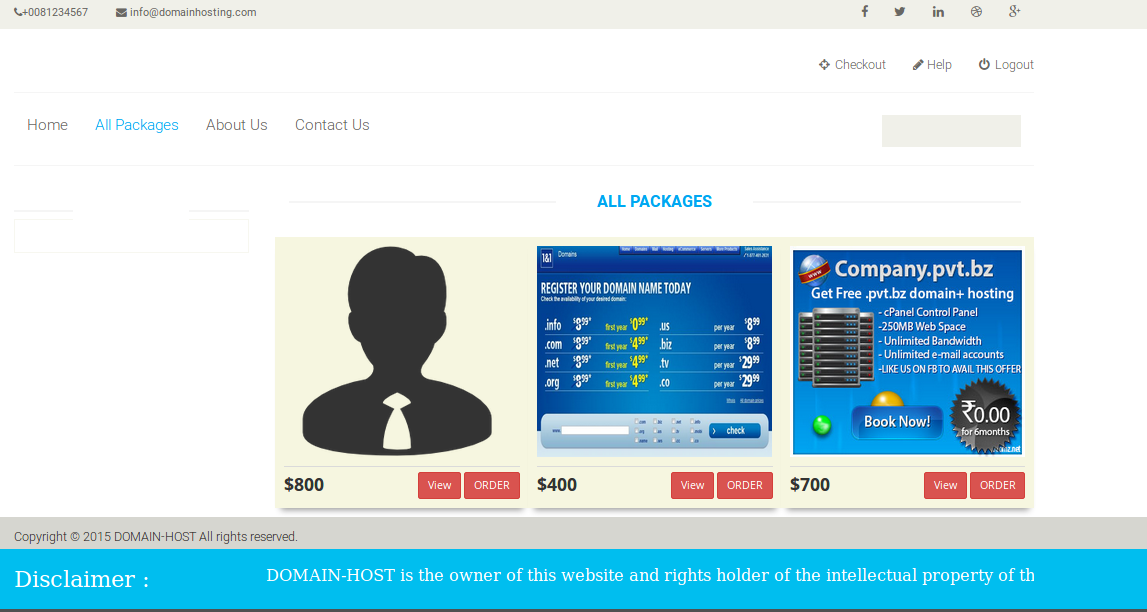
3. Dataflow of this project:

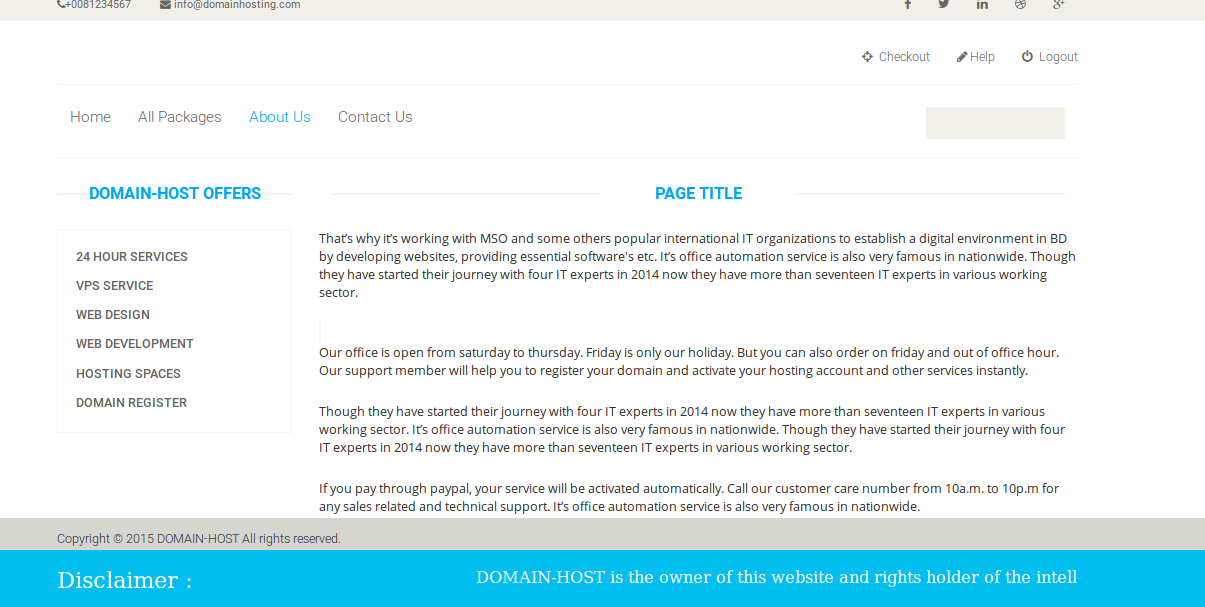


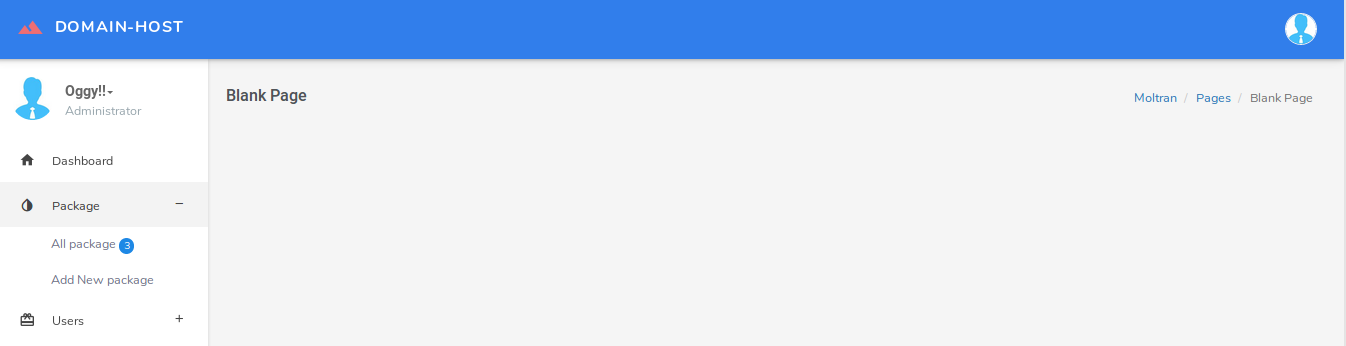
4.Screen-Shots:

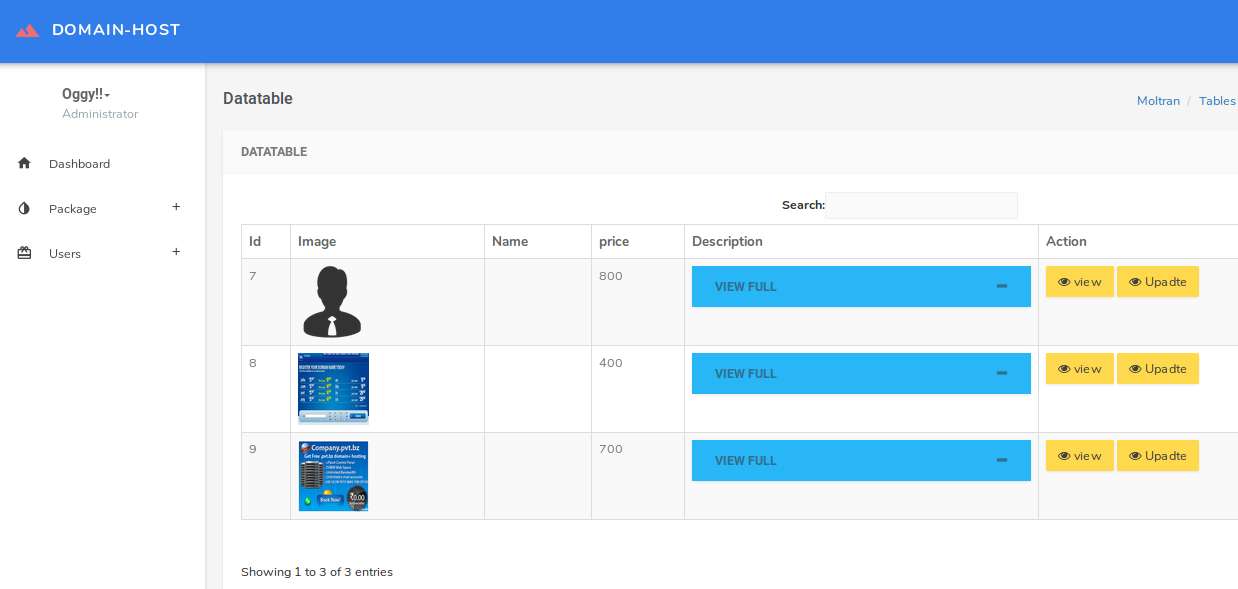
4.1 User Interface:

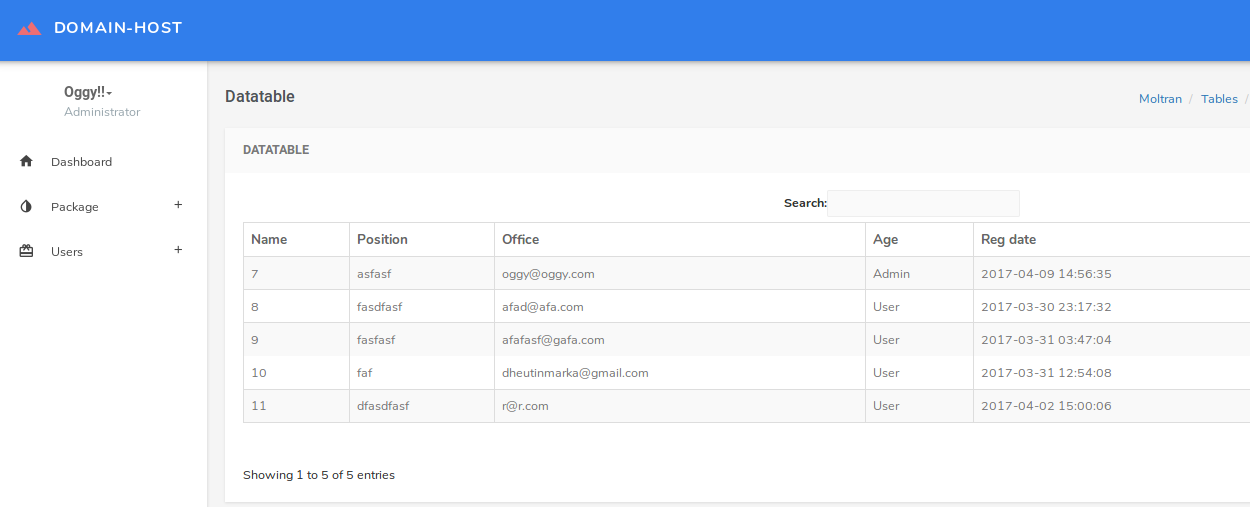


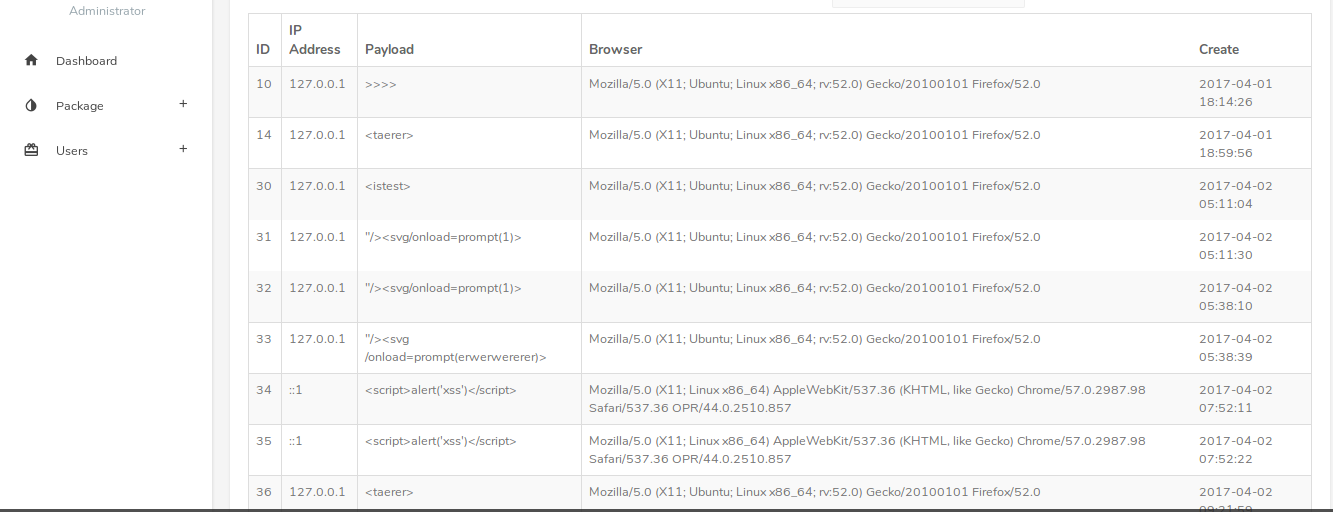


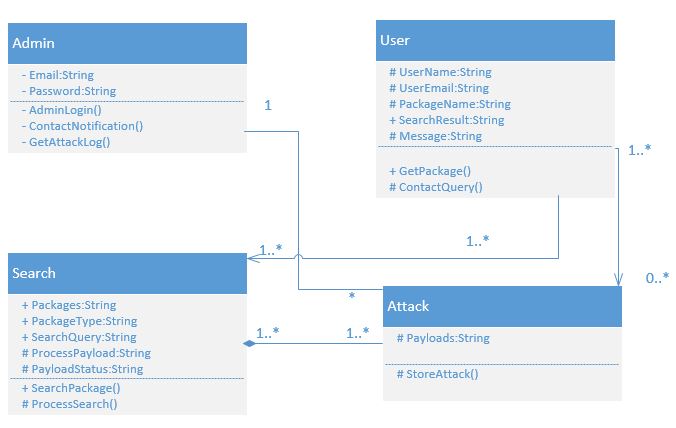


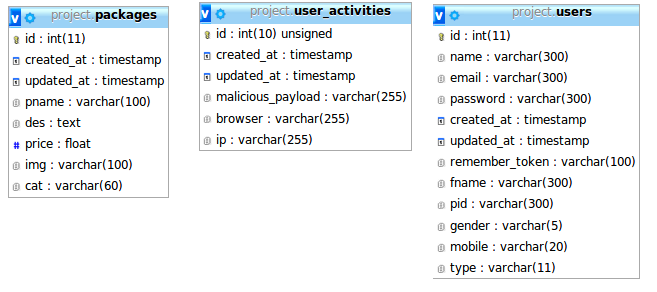
4.2 Admin Panel:





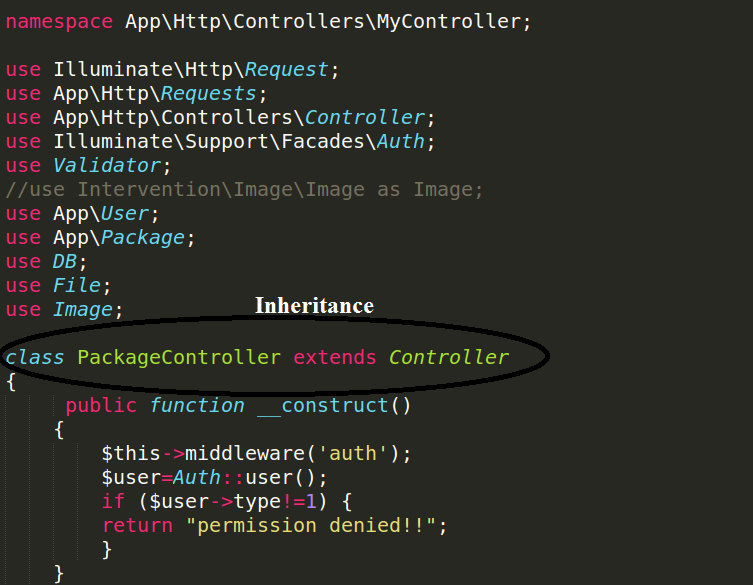


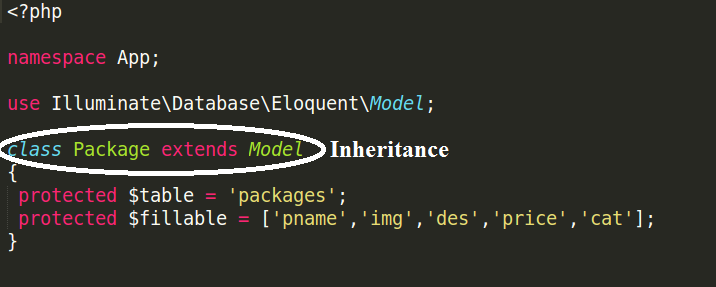
4.3 Class Diagram:

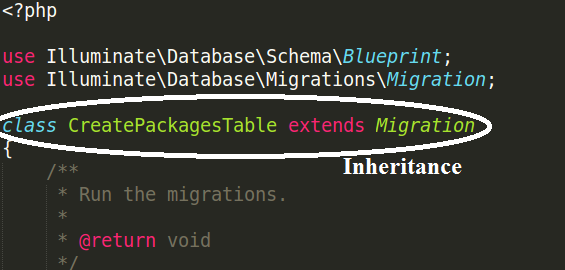
4.4 ER Diagram:

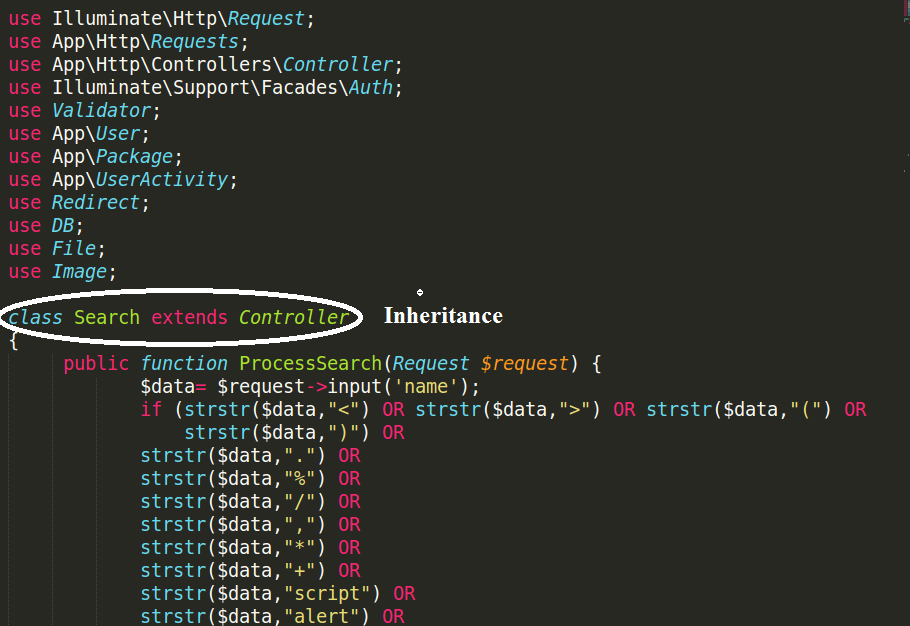
5. Coding Screen Shots:

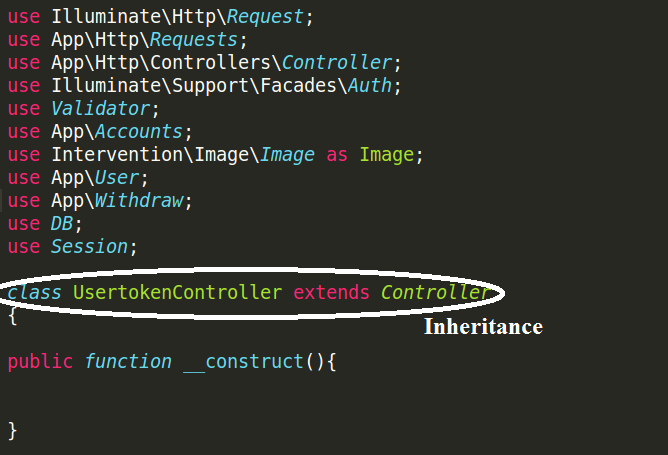
5.1 Inheritance:

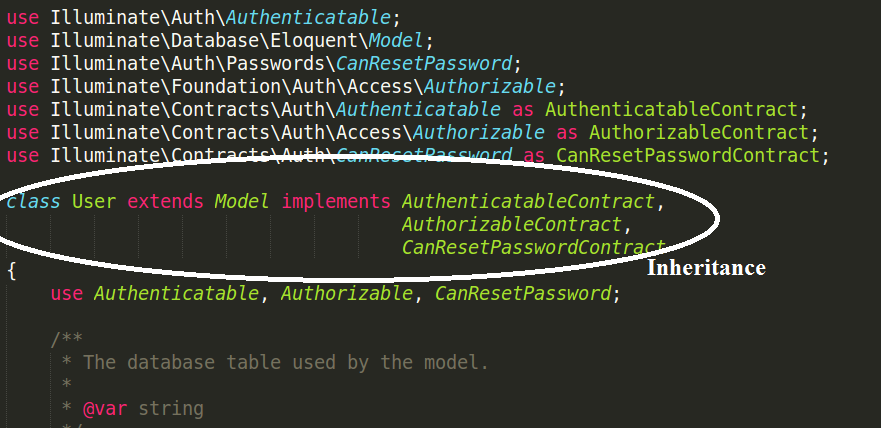


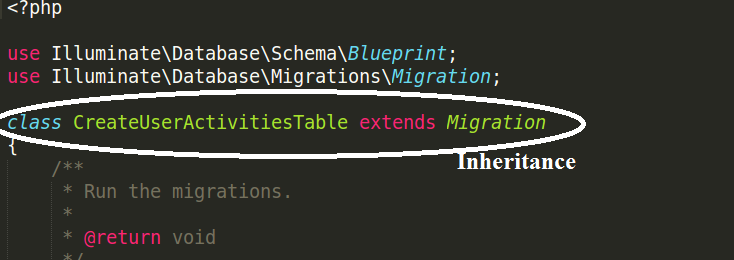


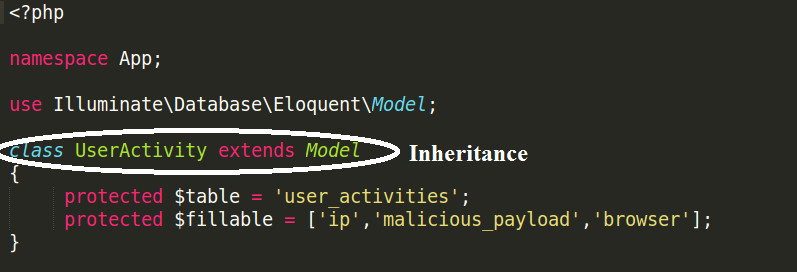




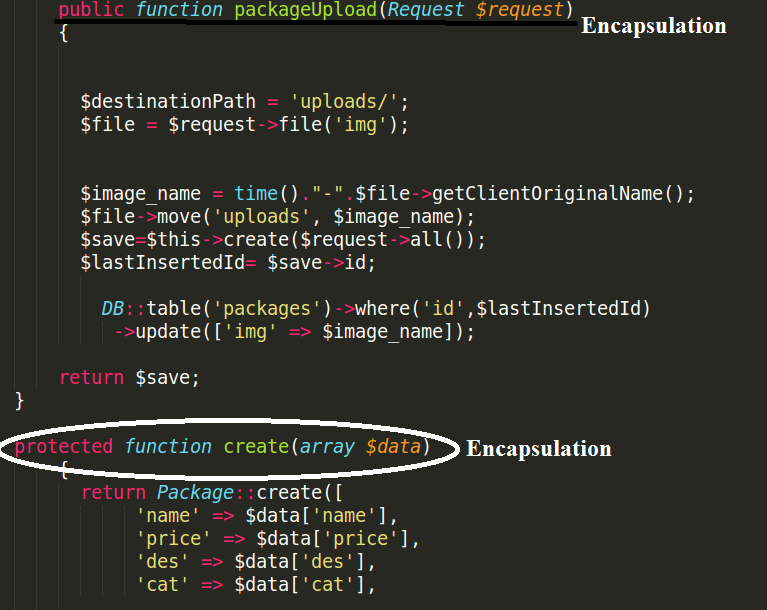


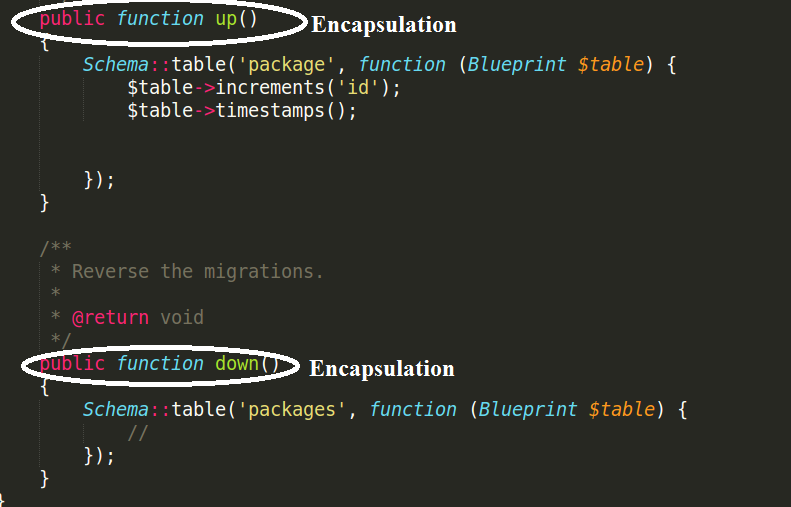


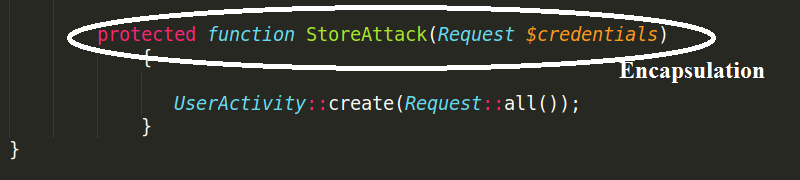


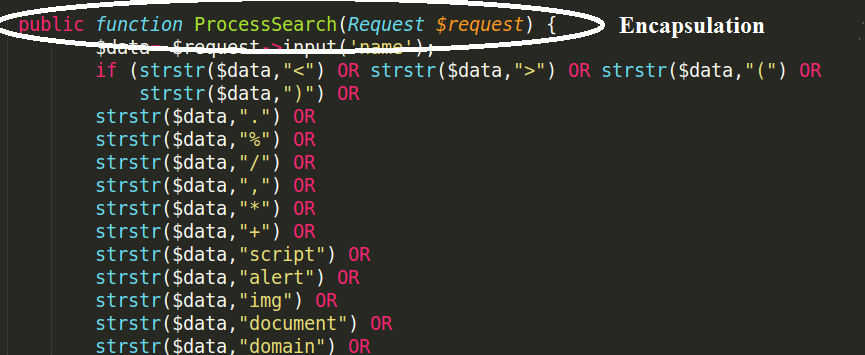


5.2 Encapsulation:

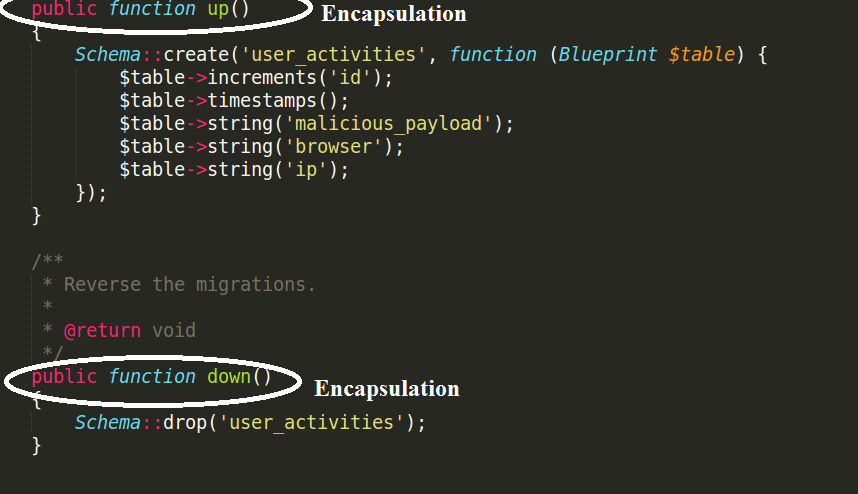


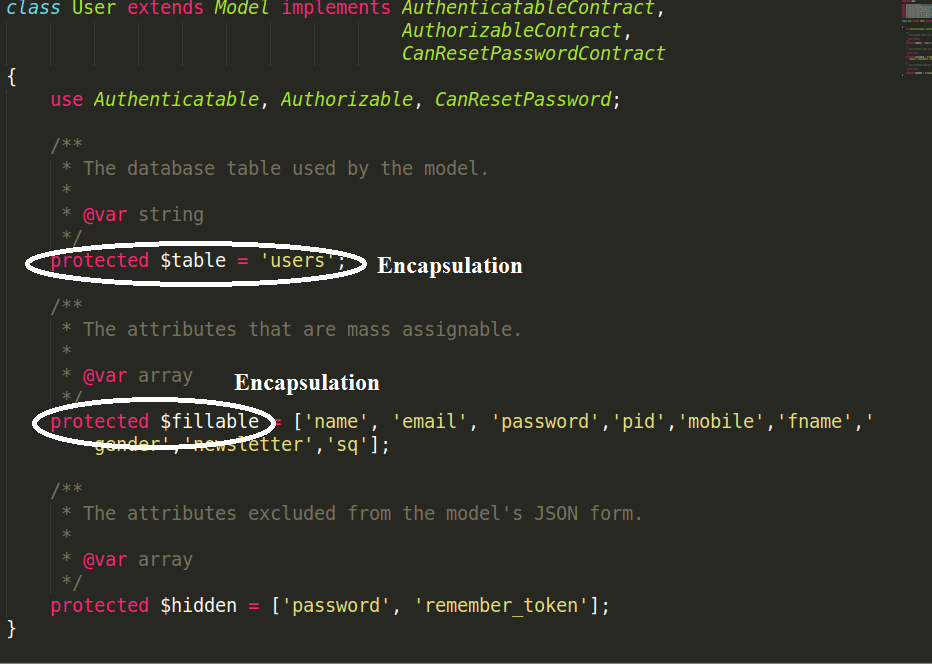












**6. Architecture:**   
Laravel application structure has an application directory called **app/** with three subdirectories: **models/,** **views/**, and **controllers/**. This is a hint that **Laravel follows the model-view-controller (MVC) architectural pattern**, which enforces a separation between “business logic” from the input and presentation logic associated with a graphical user interface (GUI). In the case of Laravel web applications, the business logic typically consists of data models for things like users, blog posts, and the GUI is just a web page in a web browser. The MVC design pattern is very popular in the web development space.

