***University of Barishal***

**Assignment on**

**Design Pattern Implementation in Web Framework**

**Course Title: Software Engineering and Information System Design**

**Course Code: CSE-3103**

**Prepared For**

Md. Erfan

Assistant Professor

Department of Computer Science & Engineering

University of Barishal

**Prepared By**

Pranta Kumar Biswas

Class Roll: 17CSE028

Session: 2016-17  
Semester: 5th

Department of Computer Science & Engineering

University of Barishal

**Problem Statement**: Here, the problem is to develop an API where we can post data and get the processed output data. The post data contains score of admission test of a student and the API will give eligible subject list as output in JSON format. For, getting any subject there will be many conditions. So, it is very difficult for us to implement too many conditions for all subjects. Design Patterns solve this problem easily. In Chain of Responsibility design pattern, we can check every subject whether it is eligible or not and check next chain. In this way, this problem can be solved. The pattern itself as the solution for the related problems, however, is represented by UML models.

**What:** Design patterns support software engineers in creating maintainable and extendable software. In order to select and apply design patterns, practitioners typically learn a pattern by reading a design pattern book or paper or by studying UML diagrams or source codes, respectively. Chain of responsibility pattern is used to achieve loose coupling in software design where a request from the client is passed to a chain of objects to process them. Later, the object in the chain will decide themselves who will be processing the request and whether the request is required to be sent to the next object in the chain or not.

**Why:** The Chain of Responsibility design pattern itself as the solution for the related problems, however, is represented by UML models.



* It reduces the coupling.
* It adds flexibility while assigning the responsibilities to objects.
* It allows a set of classes to act as one; events produced in one class can be sent to other handler classes with the help of composition.
* When more than one object can handle a request and the handler is unknown.
* When the group of objects that can handle the request must be specified in dynamic way.

These are the reason for using this design pattern in our problem.

**How:**  We need to this design pattern in Laravel Web Framework for solve the problem. At first, declare an interface and describe the methods for handling request and also decide how the client will pass the request data into the method. To eliminate duplicate boilerplate code in concrete handlers, it might be worth creating an abstract base class, derived from the interface. This class should have a field for storing a reference to the next handler in the chain. Consider making the class immutable. Now create concrete handler subclasses and implement their handling methods for each subjects. Each handler should make two decisions when receiving a request: either it’ll process the request or it’ll pass the request along the chain. After that, create instances for every subjects and make a chain with method in the Laravel controller and pass the request to the chain. This will store eligible subjects in the array list. And after the send this data in JSON format when the API is called.

**Solution Strategy:** The problem is to post exam score data to an API and the response will be eligible subject list in JSON format. To solve it with the Chain of Responsibility design pattern and Laravel Web Framework, at first we need to create a Laravel Project. Create a controller and a function for the API. Then declare the route and URL in the API of Laravel. Create a folder in the App for every class or declare every class for the design pattern in Controller. Create interface class for declaring the methods for handling requests and abstract based class for chaining subclasses. Then create subclasses for every subjects which will process the request. Every subject class will check whether it is eligible or not. If it is eligible, then it will store the name of the subject in an array. As this is call by reference, so we should take the input from the subclasses. Now in controller validate the input. Then create every instances for every subjects. Create a chain with the method declared in abstract based class and interface. Now send the request which is in the first chain. Every subject will check whether it is eligible or not and store it in an array if it is eligible for the student and send the request to the next chain. In this way an array of the eligible subject list will be implemented. And send this array as JSON format as API response. Now, anyone can use this API for an application with ajax request or other.