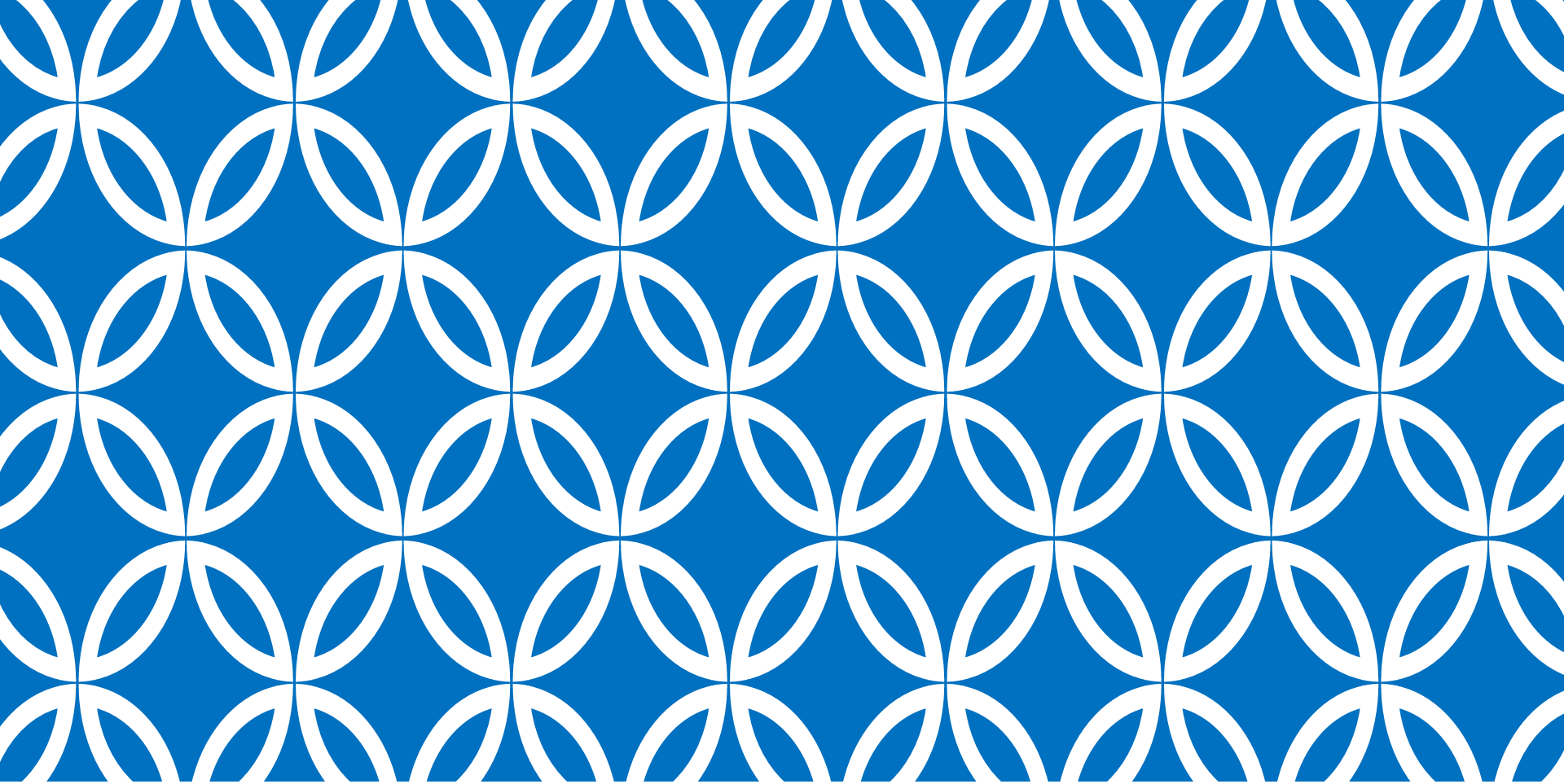
ACADGILD

FRONT END DEVELOPMENT (WITH ANGULARJS)

Website : http://www.acadgild.com
LinkedIn : https://www.linkedin.com/company/acadgild
Facebook : https://www.facebook.com/acadgild



Session 13 – AngularJS Module Configuration, Routing, Dependency Injection

Agenda – AngularJS Module Configuration, Routing, Dependency Injection

1. Exploring config and run Function For Module
2. Routing
3. Dependency Injection
4. AngularJS Service
5. AngularJS Factory
6. AngularJS Provider

Exploring config and run function for module

- We have 2 functions namely **config** and **run** function for the module.
- The config function is the first function that runs when the module loads.
- The **config** function is useful to configure providers before they are actually created.
- After the config function, the run function is executed.
- The run function can be used to perform any kind of initialization.

Routing

- AngularJS Routing helps you to divide your app into multiple views and bind different views to Controllers.
- The magic of Routing is taken care by an AngularJS service **$routeProvider**.
- **$routeProvider** service provides two methods - **.when()** and **.otherwise()** to define the routes for your app.
- Routing has dependency on **ngRoute** module.

Dependency Injection

- Dependency Injection (DI) is a software design pattern that implements inversion of control for resolving dependencies.
- AngularJS comes with a built-in dependency injection mechanism.
- You can divide your AngularJS app into multiple different types of components which AngularJS can inject into each other.
- The three ways to inject dependences are:
  - Services
  - Factories
  - Providers

AngularJS Service

- Service is a constructor function which creates the object using new keyword.
- You can add properties and functions to a service object by using this keyword.
- Unlike factory, it doesn't return anything.

Syntax:

```
app.service('serviceName',function(){})
```

When to use:

It is a singleton object. Use it when you need to share a single object across the application. **For example**, authenticated user details.

AngularJS Factory

- A factory is a simple function which allows you to add some logic before creating the object.
- It returns the created object.

Syntax:

```
app.factory('serviceName',function(){ return serviceObj;})
```

When to use:

It is just a collection of functions like a class. Hence, it can be instantiated in different controllers when you are using it with constructor function.

AngularJS Provider

- A provider is used to create a configurable service object.
- It returns value by using $get() function.

Syntax:

```
//creating a service
app.provider('serviceName',function(){});
//configuring the service
app.config(function(serviceNameProvider){});
```

When to use:

When you need to provide module-wise configuration for your service object before making it available.

Lets Discuss Assignments