# Modern Application Development – Essentials

## Objectives

After completing this course, participants will be able to explain and implement web application:

* Backend
  + API technologies overview - REST, SOAP, EJB etc
  + Scaling techniques on server side
  + Message oriented middle-ware
  + Common design patterns - MVC
* Frontend
  + HTTP and HTTP based request response cycle
  + API based web applications and how it is implemented with Servlets/JAX RS
  + Browser based client side scripting and AJAX with JQuery for frontends
  + Use of Angular JS and TDD approaches for front end
* Database - Purpose and use of NoSQL databases
* Use Test Driven development approach using Java/JUnit and Javascript based test frameworks

## Pre-Requisites

* Knowledge of Object Oriented Programming with Java
* Working exposure with any programming languages
* Recommended Java Programming Resources : Quick Intro to Java, [Java refresher with examples](http://www.cs.nmsu.edu/%257Eepontell/courses/cs272/Documents/Java_Intro/overview.pdf)
* SQL Refresher ﻿[SQL Tutorials for Beginners](https://www.codeschool.com/courses/try-sql)

**Modality, Duration** -

* 4 days Instructor Led, Hands-on.
* Delivered in two parts of 2 days each

## PART I - Course Outline - 2 Days:

Recap – Modern App Architecture, Cloud Computing and DevOps, 3-tier architecture

Backends

HTTP – Introduction, Request & response

Type of backends - API based, Dynamic Web

Dynamic Web Backends introduction and demo

Intro to RDBMS with Hibernate

### Hands on 1:

Create a little junit test to perform basic CRUD

operations on a database using hibernate

### API Backends

API backends with Servlets

Concept and foundations of REST

Comparing API styles REST and SOAP

How JAX RS makes it easy over servlets

APIs design practices

Content negotiation with REST

API Authentication, API versioning, API documentation

### Hands on 2:

Create a rest API with ALL 5 operations over the hibernate core layer created earlier

### Messaging and Middleware

Why messaging

JMS and MOM

Queues and topics

Pub sub and async sending

Messaging services (RabitmQ)

### Hands on 3:

Setup rabitMQ and write JUnit tests to perform interprocess communication over MQ

### Frontend

HTML – Basic Syntax, document structure, forms

Javascript and OO Javascript

Accessing backends with javascript

Jquery,

Selectors

DOM traversal

DOM manipulation

JQuery event emitters

AJAX and jquery

### Hands on 4:

Setup a frontend project that uses the earlier build REST api to perform an

Entity Maintenance (CRUD) usecase

### Testing

Use of camel for api testing - Junit (TDD)

Testing and dependency injection

### Hands on 5:

Write tests for the backend APIs and make sure this runs as part of the build

### Databases:

SQL pros and cons

The limitations of SQL/Relational Databases

SQL vs. NoSQL, NoSQL Flavours

Project Discussion

Project will be to create blogging app on the lines of stack overflow with the following use-cases:

* Login
* Signup
* Create a blog post
* View all blog posts
* Search blog posts

This project will be built with JQuery frontend and REST backend implemented with JAX-RS

## PART II - Course Outline 2-Days

Recap – Part 1

### NoSQL DB (Mongo db)

Setting up the database

Using the command line client to perform mongo queries

Using the java client with Morphia

### Hands on 1:

Create JUnit test cases to perform collection operations on Mongo DB using Morphia in java

### HTML5

Introduction to changes

Canvas and SVG

Server side events intro and demo

### Angular JS

Controller Component

Model Component

View Component

Directives

Filters

Services

DI in Angular.js

Routes

### Hands on 2:

Using a rest backend (Provided) create CRUD operations with Angular

Use angular routes to enhance SPA feel

### Testing

Frontend testing with Jasmine or Protractor

Testing without a backend

End to End testing with a backend

### Hands on 3:

Write test cases for angular frontend components built in previous hands on

### Evaluation Of Technologies

Objective evaluation of Java, Python

Comparison of Serverside Technologies - Node.js, Java, Python

How to scale different layers of the app: front-end

Frontend App issues and performance tips

How to scale different layers of the app: back-end

Load Balancers (HAProxy, RoundRobin, IP/RegEx based)

## Project Discussion

Project will be to enhance the project created in part 1 to use Angular fronted, Mongo DB backend and include frontend test cases as part of the build-deploy workflow