



UNIVERSITY OF RUHUNA

Faculty of Engineering

End-Semester 5, Examination in Engineering, August 2018

Module Number: EE5207

Module Name: Internet Technologies

Part - II

[2 hours]

[Answer all questions, each question carries 5 marks]

Q1. a) Explain differences between Microsoft .NET Framework and Microsoft .NET Core framework?

[1 mark]

b) Explain what is dependency injection in software development and mention advantages of using it?

[1.5 mark]

c) Explain what is a web service and mention advantages of using it?

[1 mark]

d) Explain what is Object relation mapping framework (ORM) in software development. Give two example for ORM framework.

[1.5 mark]

Q2. a) Explain Node.js Framework and mention the benefits of using it.

[1.5 marks]

b) Explain the functionality of the following code segment in Node.js.

```
const express = require('express');
const app = express();
app.get('/', function(req, res){
  res.send('Hello World');
});
app.listen(3000, function(){
  console.log('Server started on Port 3000');
});
```

[1.5 marks]

c) The following code segments are added to the Node.js program in section Q2b). Explain the functionalities of these code segments.

[2 marks]

i)

```
var bodyParser = require('body-parser');
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended:false}));

var path = require('path');
app.use(express.static(path.join(__dirname, 'public')));
```

ii)

```
app.use(function(req, res, next){
  console.log("Start");
  next();
});

app.get('/', function(req, res, next){
  res.send("Middle");
  next();
});

app.use('/', function(req, res){
  console.log('End');
});
```

iii)

```
var express = require('express');
var router = express.Router();

router.get('/get', function(req, res){
  res.send('GET route on things.');
```

```
});
router.post('/add', function(req, res){
  res.send('POST route on things.');
```

```
});

app.use('/products', router);
```

iv)

```
var exphbs = require('express-handlebars');

app.engine('handlebars', exphbs({defaultLayout: 'main'}));
app.set('view engine', 'handlebars');

app.get('/home', function (req, res) {
  res.render('home');
});
```

Q3. a) Explain the MVC design pattern using a diagram.

[1 mark]

b) Write a controller name "Product" and write the following action methods

i) Write the Greetings action for Product controller, which takes *name* and *times* as parameters and display the "Hello " + *name* + " well come" message number equal to value *times*. (You need to write Greetings View with Razor code and pass variables from controller to the View.)

ii) Explain how you can pass *name* and *times* parameters using URL.

[1 mark]

c) Create a Product class which has product id, product name, price and number in stock as properties.

[1 mark]

d) The Product class list is used to store the product details. Explain how you pass the list to the corresponding view and explain how you retrieve the list in the view.

[1 mark]

e) Explain how you would create a table to show the list of products in the View using Razor view engine.

[1 mark]

Q4. a) Create a class ConsoleLogger by implementing following C# interface. In the implementation of interface method, you have to write the message to Console.

```
public interface IWriteMessage
{
    void WriteMessage(string message);
}
```

[1 mark]

b) Using following code we can add dependencies to the ASP.NET Core. Explain the differences of the following three methods.

```
services.AddScoped<IWriteMessage, ConsoleLogger>();
services.AddTransient<IWriteMessage, ConsoleLogger>();
services.AddSingleton<IWriteMessage, ConsoleLogger>();
```

[1.5 mark]

c) In Microsoft Entity Framework Core we have several approaches to connect with the databases. Explain *Database First approach* and *Code First approach*.

[1 mark]

d) Data annotations are used to describe following Model class that represents a Book. Explain the meanings of data annotations.

[1 mark]

Listing 1: Node.js code

```
public class Book
{
    [Key]
    public int BookId { get; set; }
    [Required] [MaxLength(150)]
    public string Title { get; set; }
    [ForeignKey("Author")]
    public Author AuthorID { get; set; }
}
```

e) What is the usage of DbContext class?

[0.5 mark]

Q5. a) What is the use of Middleware in Express.js?

[1 mark]

b) Write a Middleware function which print "Logging the request" in the console for each request it receives.

[1 mark]

c) Explain the following HTTP method

- i) GET
- ii) POST
- iii) PUT
- iv) DELETE

[1 mark]

d) Create a HTML form with two input text fields to get user name and password. When the user click the submit button of the form it should send data to login method of the server.

[1 mark]

e) Write a post method to access form data and display them in the console.

[1 mark]