

Q.1

a) What is MVC pattern and its benefits?

Ans :- Model View Controller (MVC) is an architectural pattern that is used to build software applications.

→ MVC pattern works by dividing an application into three layers, which help improve modularity and reusability of the application while they are more flexible and can support iterations.

1- Model :- It carries out all the management of all the data related logic.

2- View :- User interface which is seen by the users :- Eg: HTML, CSS, JavaScript is a part of the view.

3- Controller :- Gets user-input from either a URL request or a view.

* Rest / Soap / Rest vs Soap:-

→ Rest is a type of web service - (API)

→ Application programming Interface refers to the lines of code that facilitate two different programs to engage in communication.

CSW

Q.1

- a) What is MVC pattern and its benefits?

Ans :- Model View Controller (MVC) is an architectural pattern that is used to build software applications.

→ MVC pattern works by dividing an application into three layers, which help improve modularity and reusability of the application while they are more flexible and can support iterations.

1- Model :- It carries out all the management of all the data related logic.

2- View :- User interface which is seen by the user :- Eg: HTML, CSS, JavaScript is a part of the view.

3- Controller :- Gets user-input from either a URL request or a view.

* Rest / Soap / Rest vs Soap:-

→ Rest is a type of web service - (API)

→ Application programming Interface refers to the lines of code that facilitate two different programs to engage in communication.

- RESTful API's work by decomposing a transaction into several little modules. Each of these modules is made to solve a different portion of the transaction.

SOAP → Simple Object Access Protocol

- Simple object Access Protocol (SOAP) is a messaging protocol that makes it possible for the distributed elements of an application to communicate with each other.
- SOAP uses extensible Markup Language (XML) to enable communication among web services.

Characteristics of Soap:-

- Neutrality → can function over any protocol
- Extensibility → Security & WS Addressing
- Independence → Allows any kind of programming model

REST

- REST needs a lesser amt. of bandwidth making it extremely suitable for Internet
- REST architecture does not carry any state ~~within~~ and a message

SOAP

- It is utilized in messaging systems and transport protocols
- SOAP has a very rigid structure and a message

making it suitable
for cloud based
environment

should be formatted
in XML format.

* What is ORM :-

ORM → Object Relational Mapping

ORM allows developers to access and
modify objects while it saves them
from thinking about the relation of
objects with data sources.

ORM contains details of OOP's concepts
of abstraction and details of RDBMS or
XML data sources.

* GET vs POST :- Pg 68 - BOOK PDF

GET

→ Get method is used
to transfer data from
client to server.

→ Data is transferred
via URL parameters
hence exposed to the
world

→ There is limitation
on sending data (data
size)

POST

Post method is
used to fetch data
from the server.

→ Data is transferred
via http headers
and it doesn't pass
via URL parameters

→ Post doesn't have
any limitation on
data size.

* Important elements of web development.

(*)
Elements of Web Development Include:-

1- Front-End Technologies

- i) HTML → Structural
- ii) CSS → Presentational
- iii) Javascript → Behavioral
- iv) jQuery → Javascript Library
- v) Bootstrap → CSS Library

2- BackEnd Technologies :-

- i) Java
- ii) Node JS
- iii) PHP
- iv) Python

HTML Table :-

```
<table>
  <tr>
    <th> Table Header </th>
    <th> 1 </th>
    <th> 2 </th>
  </tr>
  <tr>
    <td> table Data </td>
    <td> table Data 2 </td>
  </tr>
</table>
```

Navigation Bar

<nav>

test →

```
<ul>
  <li><a href="#">Home</a></li>
  <li><a href="-->">Services</a></li>
  <li><a href="-->">Vision
```


</nav>

Horizontal NavBar

CSS :-

ul {

list-style-type: none;

overflow: hidden;

background-color: ---;

}

li {

float: left;

li a {

display: block;

color: ---;

padding: 12px 15px;

}

li a: hover {

background-color: yellow;

}

Vertical NavBar:-

ul {

list-style-type: none;

width: 100px;

background-color: blue;
}

li a {

display: block;

color: red;

padding: 6px 12px;

li a: hover {

background-color: ...;

color: ...;

}

Dropdowns:- Question 7 (CSS Assignment)

• ~~dropdown~~ . dropdown {

background-color: ...;

color: ...;

padding: 16px;

font-size: ...;

border: none;

}

dropdown {

position: relative;

display: inline;

}

dropdown-content {

```
display: none;  
position: absolute;  
background-color: #f1f1f1;  
max-width: 200px;  
z-index: 1.  
}
```

• dropdown-content a {

```
display: block;  
text-decoration: none;
```

{

• dropdown-content a:hover {

```
background-color: #ccc;  
}
```

dropdown: hover → dropdown-content {
display: block
}

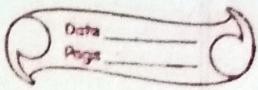
• dropdown: hover, & drop bth { background-color: #ccc;

HTML Assignment :- Q2 R3

Types of selector:-

- Element selector
- . class selector
- #id selector

BOX MODEL → pg 122 - PDF



JQuery

Hide & Show :-

```
$ (document).ready(function() {  
    $("#a").click(function() {  
        $("p").hide();  
    });  
  
    $("#b").click(function() {  
        $("p").show();  
    });  
});
```

Toggle :-

```
$ (document).ready(function() {  
    $("button").click(function() {  
        $("a").toggle();  
    });  
});
```

Animate :-

```
$ (document).ready(function() {  
    $("button").click(function() {  
        $("div").animate({ left: '500px' },  
    });  
});
```

Opacity

height

width

width: toggle ...

dropdown:- HTML

```
<div class="dropdown">  
  <button class="dropbtn"> Dropdown </button>  
  <div class="dropdown-content">  
    <a href="#"> Link 1 </a>  
    <a href="#"> Link 2 </a>  
    <a href="#"> Link 3 </a>  
  </div>  
</div>  
</body>  
</html>
```

Bon Model :-

- Margins
- Borders
- Padding
- Content :- text/images

Style

div {

background-color: #fff;

border: 2px solid black;

padding: 5px;

margin: 5px;

}

HTML

```
<div> content / text / image  
</div>
```

DOM Manipulation :-

DOM is a standard which is used to access HTML & XML documents.

- 1- n.innerHTML →

Eg:- ~~~~ Vivek Mohanta

a. innerHTML will give → Vivek
Mohanta.

- 2- n.nodeName :- gives the name of the node 'n'.

- 3- n.nodeValue :- gives the inner text of node 'n'. It does not parse HTML & gets value in text format, faster than innerHTML.

- 4- n.parentNode :- Gives access to the parent node of 'n'

- 5- n.childNodes :- gives access to the child nodes of 'n'

- 6- n.attributes :- gives all the attributes of 'n'.

Focus :-BOOTS TRAP :-

→ Pagination :-

```
<ul class="pagination">
  <li class="page-item"><a class="page-link"
    href="#">Last </a></li>
  <li class="page-item"><a class="page-link"
    href="#">1 </a></li>
  ...
  ...
</ul>
```

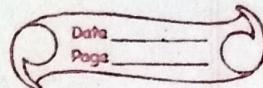
```
<li class="page-item"><a class="page-link"
  href="#">Next </a></li>
</ul>
```

Active → <li class="page-item active">

→ Responsive grid :-

```
> <div class="row">
  <div class="col"> column 1 </div>
  <div class="col"> column 2 </div>
  <div class="col"> column 3 </div>
  <div class="col-md-3"> column 4 </div>
</div>
```

Column of medium size taking
3 units of space. [Total space = 12]



→ Progress-Bar :-

```
<div class="progress">
  <div class="progress-bar" style="width: 50%;">
    50% </div>
</div>
```

→ <div class="progress-bar progress-bar-striped progress-bar-animated" style="width: 70%;> Loading ... </div>

- bg-dark → black
- bg-danger → red
- bg-success → green
- bg-primary → blue
- bg-info → lightblue
- bg-warning → yellow

→ Dropdown :-

```
<div class="btn-group">
  <button class="btn btn-primary dropdown-toggle" data-toggle="dropdown" id="dropdownMenuButton">
    Item </button>
  <div class="dropdown-menu">
    <a class="dropdown-item" href="#">Lorema</a>
    <a class="dropdown-item" href="#">...</a>
    <a class="dropdown-item" href="#">...</a>
  </div>
</div>
```

NavBar :-

```

<nav class="navbar navbar-expand-sm
bg-primary navbar-dark">
  <ul class="navbar-nav">
    <li class="nav-item active">
      <a class="nav-link href="#"> Home </a>
    </li>
    <li class="nav-item">
      <!--
      </li>
    <li class="nav-item">
      <a class="nav-link disabled" href="#">
        careers </a>
    </li>
  </ul>
</nav>

```

Form :-

```

<div class="container">
  <form>
    <div class="form-group">
      <label for="email"> Email Address </label>
      <input type="email" class="form-control"
        id="email" placeholder="-----" />
    </div>
    <div class="form-group">
      <label for="password"> Password: </label>
      <input type="password" class="form-control"
        placeholder="Type your password here -" />
    </div>
  </form>

```

Carousel :-

<style>

```
• carousel-inner img {  
    width: 100%;  
    height: 100%;  
}
```

</style>

```
<div id="carouselExampleControls" class="carousel slide"  
      data-slide="carousel">  
  <div class="carousel-inner">  
    <div class="carousel-item active">  
        
    </div>  
    <div class="carousel-item">  
        
    </div>  
    <div class="carousel-item">  
        
    </div>  
  </div>  
  <a class="carousel-control-prev" href="#carousel  
      ExampleControls" role="button" data-slide="prev">  
    <span class="carousel-control-prev-icon"  
        aria-hidden="true"></span>  
    <span class="sr-only">Previous</span>  
  </a>  
  <a class="carousel-control-next" href="#" data-slide="next"  
      data-slide="next" role="button" data-slide="next">  
    <span class="carousel-control-next-icon"  
        aria-hidden="true"></span>  
  </a>
```

Popup Box without using javascript alert.

<script>

function myFunction()

var txt;

if (confirm("Press a button")) {

txt = "You pressed OK!";

} else {

txt = "You pressed cancel!";

}

document.getElementById("demo").innerHTML

=txt;

}

</script>

<body>

<h2> Javascript Alert Box <h2>

<button onclick="myFunction ()"> Try It

</button>

<p id="demo"></p>

</body>