# Bootstrap, React, Node.js, and MongoDB Programs – Beginner Guide

## Q1

### a. Bootstrap 5 - Responsive Layout

<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="UTF-8">  
<meta name="viewport" content="width=device-width, initial-scale=1">  
<title>Bootstrap Container Example</title>  
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">  
</head>  
<body>  
  
<div class="container">  
 <h2>.container Example</h2>  
 <p>This container has fixed width. It adjusts based on screen size.</p>  
</div>  
  
<div class="container-fluid">  
 <h2>.container-fluid Example</h2>  
 <p>This container is full width and takes up 100% of the screen.</p>  
</div>  
  
</body>  
</html>

### b. React - Simple To-Do List App

1.b.To-Do list:

import React, { useState } from "react";

function App() {

const [task, setTask] = useState("");

const [list, setList] = useState([]);

const addTask = () => {

setList([...list,{text:task,done:false}]);

setTask("");

};

const toggleTask = (index) => {

const newList = [...list];

newList[index].done = !newList[index].done;

setList(newList);

};

const deleteTask = (index) => {

const newList = list.filter((\_, i) => i !== index);

setList(newList);

};

return (

<div className="container mt-4">

<h3>Simple To-Do List</h3>

<input

value={task}

onChange={(e) => setTask(e.target.value)}

placeholder="Enter task"

className="form-control mb-2"

/>

<button className="btn btn-primary mb-3" onClick={addTask}>

Add Task

</button>

<ul className="list-group">

{list.map((item, index) => (

<li

key={index}

className="list-group-item d-flex justify-content-between align-items-center"

>

<span

style={{ textDecoration: item.done ? "line-through" : "none" }}

onClick={() => toggleTask(index)}

>

{item.text}

</span>

<button

className="btn btn-danger btn-sm"

onClick={() => deleteTask(index)}

>

Delete

</button>

</li>

))}

</ul>

</div>

);

}

export default App;

Steps to run React app:

1. npx create-react-app todo-app  
2. cd todo-app  
3. Replace src/App.js with above code  
4. npm start  
5. Open http://localhost:3000 in browser

## Q2

### a. Bootstrap - Student Details Table

<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="UTF-8">  
<meta name="viewport" content="width=device-width, initial-scale=1">  
<title>Student Table</title>  
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">  
</head>  
<body>  
  
<div class="container">  
<h2>Student Details</h2>  
<table class="table table-striped table-bordered">  
<thead>  
<tr>  
<th>Name</th>  
<th>Roll No</th>  
<th>Department</th>  
<th>Email</th>  
</tr>  
</thead>  
<tbody>  
<tr><td>Ravi Kumar</td><td>101</td><td>CSE</td><td>ravi@example.com</td></tr>  
<tr><td>Anjali Singh</td><td>102</td><td>ECE</td><td>anjali@example.com</td></tr>  
<tr><td>Deepak Verma</td><td>103</td><td>IT</td><td>deepak@example.com</td></tr>  
</tbody>  
</table>  
</div>  
  
</body>  
</html>

### b. React - Controlled Signup Form

import React, { useState } from "react";  
  
function App() {  
 const [formData, setFormData] = useState({ name:"", email:"", password:"" });  
  
 function handleChange(e) {  
 const { name, value } = e.target;  
 setFormData({ ...formData, [name]: value });  
 }  
  
 function handleSubmit(e) {  
 e.preventDefault();  
 alert(`Name: ${formData.name}\nEmail: ${formData.email}`);  
 }  
  
 return (  
 <div>  
 <h2>Signup Form</h2>  
 <form onSubmit={handleSubmit}>  
 <label>Name:</label><br/>  
 <input type="text" name="name" value={formData.name} onChange={handleChange}/><br/>  
 <label>Email:</label><br/>  
 <input type="email" name="email" value={formData.email} onChange={handleChange}/><br/>  
 <label>Password:</label><br/>  
 <input type="password" name="password" value={formData.password} onChange={handleChange}/><br/>  
 <button type="submit">Register</button>  
 </form>  
 </div>  
 );  
}  
  
export default App;

Steps to run React Signup Form:

1. npx create-react-app signup-form  
2. cd signup-form  
3. Replace src/App.js with above code  
4. npm start  
5. Open http://localhost:3000 in browser

Explanation: useState stores form data, handleChange updates state, handleSubmit shows alert, each input is controlled by React.

## Q3

### a. Bootstrap - Buttons Example

<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="UTF-8">  
<meta name="viewport" content="width=device-width, initial-scale=1">  
<title>Bootstrap Buttons</title>  
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">  
</head>  
<body>  
  
<div class="container mt-3">  
<button class="btn btn-primary">Primary</button>  
<button class="btn btn-success">Success</button>  
<button class="btn btn-warning">Warning</button>  
<button class="btn btn-outline-danger">Danger</button>  
</div>  
  
</body>  
</html>

### b. React - User Authentication Form

import React, { useState } from "react";  
  
function AuthForm() {  
 const [form, setForm] = useState({ username:"", email:"", password:"" });  
  
 function handleChange(e) {  
 const { name, value } = e.target;  
 setForm({ ...form, [name]: value });  
 }  
  
 function handleSubmit(e) {  
 e.preventDefault();  
 alert(`Username: ${form.username}\nEmail: ${form.email}`);  
 }  
  
 return (  
 <div>  
 <h2>Login / Registration Form</h2>  
 <form onSubmit={handleSubmit}>  
 <label>Username:</label><br/>  
 <input name="username" value={form.username} onChange={handleChange}/><br/>  
 <label>Email:</label><br/>  
 <input name="email" value={form.email} onChange={handleChange}/><br/>  
 <label>Password:</label><br/>  
 <input type="password" name="password" value={form.password} onChange={handleChange}/><br/>  
 <button type="submit">Submit</button>  
 </form>  
 </div>  
 );  
}  
  
export default AuthForm;

Steps to run React Authentication Form: same as previous React apps, replace App.js with above code.

## Q4

### a. Bootstrap - Dismissible Alert

<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="UTF-8">  
<meta name="viewport" content="width=device-width, initial-scale=1">  
<title>Bootstrap Alert</title>  
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">  
</head>  
<body>  
  
<div class="container mt-3">  
<div class="alert alert-success alert-dismissible fade show" role="alert">  
Form submitted successfully!  
<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>  
</div>  
</div>  
  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>  
</body>  
</html>

### b. Node.js / MongoDB - Employee Payroll CRUD Example

// File: server.js  
const express = require('express');  
const bodyParser = require('body-parser');  
const mongoose = require('mongoose');  
  
const app = express();  
app.use(bodyParser.json());  
  
mongoose.connect('mongodb://localhost:27017/payrollDB', { useNewUrlParser: true, useUnifiedTopology: true });  
  
const employeeSchema = new mongoose.Schema({  
 name: String,  
 position: String,  
 salary: Number  
});  
const Employee = mongoose.model('Employee', employeeSchema);  
  
// Create employee  
app.post('/employees', async (req, res) => {  
 const emp = new Employee(req.body);  
 await emp.save();  
 res.send('Employee added');  
});  
  
// Read employees  
app.get('/employees', async (req, res) => {  
 const emps = await Employee.find();  
 res.send(emps);  
});  
  
// Update employee  
app.put('/employees/:id', async (req, res) => {  
 await Employee.findByIdAndUpdate(req.params.id, req.body);  
 res.send('Employee updated');  
});  
  
// Delete employee  
app.delete('/employees/:id', async (req, res) => {  
 await Employee.findByIdAndDelete(req.params.id);  
 res.send('Employee deleted');  
});  
  
app.listen(3000, () => console.log('Server running on http://localhost:3000'));

Steps to Run:  
1. npm init -y  
2. npm install express mongoose body-parser  
3. node server.js  
4. Use Postman or browser to perform CRUD operations at http://localhost:3000/employees

## Q5

### a. Bootstrap - Responsive Table

<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="UTF-8">  
<meta name="viewport" content="width=device-width, initial-scale=1">  
<title>Responsive Table</title>  
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">  
</head>  
<body>  
  
<div class="container mt-3">  
<table class="table table-bordered table-striped table-responsive">  
<thead>  
<tr><th>Item</th><th>Quantity</th><th>Price</th></tr>  
</thead>  
<tbody>  
<tr><td>Pen</td><td>20</td><td>50</td></tr>  
<tr><td>Notebook</td><td>10</td><td>100</td></tr>  
<tr><td>Bag</td><td>5</td><td>500</td></tr>  
</tbody>  
</table>  
</div>  
</body>  
</html>

### b. Node.js - Calculator Module

// File: calculator.js  
function add(a, b) { return a + b; }  
function subtract(a, b) { return a - b; }  
function multiply(a, b) { return a \* b; }  
function divide(a, b) { return a / b; }  
module.exports = { add, subtract, multiply, divide };  
  
// File: app.js  
const calc = require('./calculator');  
console.log('Add:', calc.add(5, 3));  
console.log('Subtract:', calc.subtract(5, 3));  
console.log('Multiply:', calc.multiply(5, 3));  
console.log('Divide:', calc.divide(5, 3));

Steps to run Node.js calculator:  
1. node app.js  
2. Output shows calculation results in console

## Q6

### a. Bootstrap - School Web Page with Navbar

<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="UTF-8">  
<meta name="viewport" content="width=device-width, initial-scale=1">  
<title>School Web Page</title>  
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">  
</head>  
<body>  
  
<nav class="navbar navbar-expand-lg navbar-dark bg-primary">  
 <div class="container-fluid">  
 <a class="navbar-brand" href="#">School</a>  
 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">  
 <span class="navbar-toggler-icon"></span>  
 </button>  
 <div class="collapse navbar-collapse" id="navbarNav">  
 <ul class="navbar-nav">  
 <li class="nav-item"><a class="nav-link active" href="#">Home</a></li>  
 <li class="nav-item"><a class="nav-link" href="#">About</a></li>  
 <li class="nav-item"><a class="nav-link" href="#">Academics</a></li>  
 <li class="nav-item"><a class="nav-link" href="#">Admissions</a></li>  
 <li class="nav-item"><a class="nav-link" href="#">Contact</a></li>  
 </ul>  
 </div>  
 </div>  
</nav>  
  
<div class="container mt-3">  
<h2>Welcome to Our School</h2>  
<p>This is a simple responsive school web page with navigation links.</p>  
</div>  
  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>  
</body>  
</html>

### b. Node.js - Registration Page (GET & POST)

// File: app.js  
const express = require('express');  
const bodyParser = require('body-parser');  
const app = express();  
  
app.use(bodyParser.urlencoded({ extended: true }));  
  
app.get('/register', (req, res) => {  
 res.send(`<form method="POST">  
 Username: <input name="username"/><br/>  
 Email: <input name="email"/><br/>  
 <button type="submit">Submit</button>  
 </form>`);  
});  
  
app.post('/register', (req, res) => {  
 res.send(`User Registered:<br/>Username: ${req.body.username}<br/>Email: ${req.body.email}`);  
});  
  
app.listen(3000, () => console.log('Server running on http://localhost:3000'));

Steps: node app.js, visit http://localhost:3000/register

## Q7

### a. Node.js - College Website Server

const express = require('express');  
const app = express();  
  
app.get('/courses', (req, res) => res.send('Courses List'));  
app.get('/departments', (req, res) => res.send('Departments List'));  
  
app.listen(3000, () => console.log('Server running on http://localhost:3000'));

### b. Bootstrap - Button Group

<div class="btn-group" role="group">  
 <button type="button" class="btn btn-primary">Left</button>  
 <button type="button" class="btn btn-secondary">Middle</button>  
 <button type="button" class="btn btn-success">Right</button>  
</div>

## Q8

### a. Bootstrap Navbar with Dropdown

<nav class="navbar navbar-expand-lg navbar-light bg-light">  
 <div class="container-fluid">  
 <a class="navbar-brand" href="#">Portfolio</a>  
 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">  
 <span class="navbar-toggler-icon"></span>  
 </button>  
 <div class="collapse navbar-collapse" id="navbarNav">  
 <ul class="navbar-nav">  
 <li class="nav-item"><a class="nav-link" href="#">Home</a></li>  
 <li class="nav-item"><a class="nav-link" href="#">About</a></li>  
 <li class="nav-item dropdown">  
 <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown">Projects</a>  
 <ul class="dropdown-menu">  
 <li><a class="dropdown-item" href="#">Web</a></li>  
 <li><a class="dropdown-item" href="#">Mobile</a></li>  
 <li><a class="dropdown-item" href="#">AI</a></li>  
 </ul>  
 </li>  
 <li class="nav-item"><a class="nav-link" href="#">Contact</a></li>  
 </ul>  
 </div>  
 </div>  
</nav>  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

### b. MongoDB - Student Details

// Mongo Shell Commands  
use schoolDB  
db.students.insertMany([  
 {name:"Ravi", GPA:9},  
 {name:"Anjali", GPA:8.5},  
 {name:"Deepak", GPA:7.8}  
])  
// Retrieve students with GPA > 8  
db.students.find({GPA: {$gt: 8}})  
// Display max GPA  
db.students.find().sort({GPA:-1}).limit(1)

## Q9

### a. MongoDB - Library Book Inventory

// Mongo Shell Commands  
use libraryDB  
db.books.insertMany([  
 {title:"Book1", author:"Author1", genre:"Mystery", year:2010, available:true, copies:5},  
 {title:"Book2", author:"Author2", genre:"Thriller", year:2015, available:true, copies:3},  
 {title:"Book3", author:"Author3", genre:"Romance", year:2005, available:false, copies:2},  
 {title:"Book4", author:"Author4", genre:"Mystery", year:2020, available:true, copies:4},  
 {title:"Book5", author:"Author5", genre:"Fiction", year:1998, available:true, copies:6}  
])  
// Retrieve Mystery or Thriller books  
db.books.find({genre: {$in:["Mystery","Thriller"]}})  
// Delete books before 2000  
db.books.deleteMany({year: {$lt:2000}})

### b. Node.js - Online Shopping Web Server

const express = require('express');  
const app = express();  
  
app.get('/products', (req,res)=>res.send('Products List'));  
app.get('/contact', (req,res)=>res.send('Contact Page'));  
  
app.listen(5000, ()=>console.log('Server running on http://localhost:5000'));

## Q10

### a. Bootstrap - Registration Form with Validation

<form class="needs-validation" novalidate>  
 <div class="mb-3">  
 <label>Username</label>  
 <input type="text" class="form-control" required>  
 <div class="invalid-feedback">Please enter username.</div>  
 </div>  
 <div class="mb-3">  
 <label>Email</label>  
 <input type="email" class="form-control" required>  
 <div class="invalid-feedback">Enter valid email.</div>  
 </div>  
 <div class="mb-3">  
 <label>Password</label>  
 <input type="password" class="form-control" required>  
 <div class="invalid-feedback">Enter password.</div>  
 </div>  
 <div class="mb-3">  
 <label>Phone</label>  
 <input type="tel" class="form-control" required>  
 <div class="invalid-feedback">Enter phone number.</div>  
 </div>  
 <button class="btn btn-primary" type="submit">Register</button>  
</form>  
<script>  
(() => {  
 'use strict'  
 const forms = document.querySelectorAll('.needs-validation')  
 Array.from(forms).forEach(form => {  
 form.addEventListener('submit', event => {  
 if (!form.checkValidity()) {  
 event.preventDefault()  
 event.stopPropagation()  
 }  
 form.classList.add('was-validated')  
 }, false)  
 })  
})()  
</script>

### b. MongoDB - Employee Details

// Mongo Shell Commands  
use companyDB  
db.employee.insertMany([  
 {name:"Ravi", designation:"Manager", address:"A Street", salary:40000},  
 {name:"Anjali", designation:"Developer", address:"B Street", salary:35000},  
 {name:"Deepak", designation:"Tester", address:"C Street", salary:28000},  
 {name:"Sita", designation:"HR", address:"D Street", salary:30000},  
 {name:"Rohan", designation:"Designer", address:"E Street", salary:45000}  
])  
// Retrieve salary > 30000  
db.employee.find({salary: {$gt: 30000}})  
// Update address of two employees  
db.employee.updateOne({name:"Ravi"}, {$set:{address:"New A Street"}})  
db.employee.updateOne({name:"Anjali"}, {$set:{address:"New B Street"}})