Simple Answers — Questions 1 to 10

1.a Bootstrap: .container vs .container-fluid

```
<div class="container bg-light p-3 mb-3 border"> <h2>Fixed .container</h2> <p>Content inside .container is centered with fixed max-width at breakpoints.</p> </div> <div class="container-fluid bg-info text-white p-3 border"> <h2>Full-width .container-fluid</h2> <p>Content spans the full width of the viewport.</p> </div>
```

1.b Simple React To-Do (localStorage + Hooks)

```
import React, {useState, useEffect} from 'react'; function TodoApp(){ const [tasks, setTasks] = useState(()=>JSON.parse(localStorage.getItem('tasks'))||[]); const [title, setTitle] = useState(''); useEffect(()=>{ localStorage.setItem('tasks', JSON.stringify(tasks)); }, [tasks]); const addTask = () => { if(!title) return; setTasks([{title, complete:false}, ...tasks]); setTitle(''); }; const toggle = i => { const t=tasks.slice(); t[i].complete=!t[i].complete; setTasks(t); }; const remove = i => { setTasks(tasks.filter((_,idx)=>idx!==i)); }; return ( <div> <input value={title} onChange={e=>setTitle(e.target.value)} /> <button onClick={addTask}>Add</button> <ul>{tasks.map((t,i)=><li key={i}> <span style={{textDecoration: t.complete?'line-through':''}}>{t.title}</span> <button onClick={()=>toggle(i)}>Toggle</button> <button onClick={()=>remove(i)}>Delete</button> </li>)}</ul> </div> ); } export default TodoApp;
```

2.a Student Details Table (Bootstrap)

```
<table class="table table-striped table-bordered"> <thead class="table-dark">
<tr><th>Name</th><th>Roll No</th><th>Department</th><th>Email</th></tr> </thead> <tbody>
<tr><td>Alice</td><td>101</td><td>IT</td><td>alice@uni.edu</td></tr>
<tr><td>Bob</td><td>102</td><td>CSE</td><td>bob@uni.edu</td></tr> </tbody> </table>
```

2.b Controlled Signup Form (React)

```
import React, {useState} from 'react'; function Signup(){ const [form, setForm] =
useState({name:'', email:'', password:''}); const onChange = e => setForm({...form,
[e.target.name]: e.target.value}); const onSubmit = e => { e.preventDefault(); alert('Registered:
'+form.name); }; return ( <form onSubmit={onSubmit}> <input name="name" value={form.name}
onChange={onChange} placeholder="Name" /> <input name="email" value={form.email}
onChange={onChange} placeholder="Email" /> <input name="password" type="password"
value={form.password} onChange={onChange} placeholder="Password" /> <button type="submit">Sign
Up</button> </form> ); } export default Signup;
```

3.a Bootstrap Buttons

```
<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">Outline
Danger</button>
```

3.b React Simple Auth (client-side only)

```
import React, {useState} from 'react'; function Auth(){ const
[user,setUser]=useState({username:'',password:''}); const
onChange=e=>setUser({...user,[e.target.name]:e.target.value}); const register=e=>{
e.preventDefault(); alert('Registered '+user.username); }; return ( <form onSubmit={register}>
<input name="username" onChange={onChange} placeholder="Username" /> <input name="password"
onChange={onChange} placeholder="Password" type="password" /> <button>Register</button> </form>
); } export default Auth;
```

4.a Dismissible Alert (Bootstrap)

```
<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>
```

4.b Employee Payroll - MongoDB CRUD (simple commands)

```
// insert db.employees.insertOne({name:'John', designation:'Developer', address:'Delhi',
salary:50000}); // read db.employees.find(); // update db.employees.updateOne({name:'John'},
{$set:{address:'Mumbai'}}); // delete db.employees.deleteOne({name:'John'});
```

5.a Responsive Table Example

```
<div class="table-responsive"> <table class="table table-striped table-bordered">
<thead><tr><th>Item</th><th>Category</th><th>Price</th><th>Stock</th></tr></thead> <tbody>
<tr><td>Fabric</td><td>Raw</td><td>1200</td><td>50</td></tr> </tbody> </table> </div>
```

5.b Calculator Module (Node.js)

```
// calculator.js exports.add = (a,b)=>a+b; exports.sub = (a,b)=>a-b; exports.mul = (a,b)=>a*b;
exports.div = (a,b)=> b===0? null : a/b; // app.js const calc = require('./calculator');
console.log(calc.add(2,3));
```

6.a School Webpage Navbar (Bootstrap)

```
<nav class="navbar navbar-expand-lg navbar-dark bg-primary"> <div class="container-fluid"> <a
class="navbar-brand" href="#">My School</a> <button class="navbar-toggler"
data-bs-toggle="collapse" data-bs-target="#nav"> <span class="navbar-toggler-icon"></span>
</button> <div class="collapse navbar-collapse" id="nav"> <ul class="navbar-nav ms-auto"> <li
class="nav-item"><a class="nav-link" href="#home">Home</a></li> <li class="nav-item"><a
class="nav-link" href="#about">About</a></li> <li class="nav-item"><a class="nav-link"
href="#academics">Academics</a></li> <li class="nav-item"><a class="nav-link"
href="#admissions">Admissions</a></li> <li class="nav-item"><a class="nav-link"
href="#contact">Contact</a></li> </ul> </div> </div> </nav>
```

6.b Registration (GET & POST) - Node.js (Express)

```
const express = require('express'); const app = express();
app.use(express.urlencoded({extended:true})); app.get('/register', (req,res)=>{ res.send('<form
method="POST"><input name="name"/><button>Submit</button></form>'); }); app.post('/register',
(req,res)=>{ res.send('Thanks '+req.body.name); }); app.listen(3000);
```

7.a HTTP Server with /courses and /departments (Node)

```
const http = require('http'); const server = http.createServer((req,res)=>{ if(req.url === '/courses') res.end(JSON.stringify(['BCA','B.Tech','MCA'])); else if(req.url === '/departments') res.end(JSON.stringify(['IT','CSE','ECE'])); else res.end('College Website'); }); server.listen(3000);
```

7.b Button Group (Bootstrap)

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

8.a Navbar with Projects Dropdown (Bootstrap)

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark"> <div class="container-fluid"> <a
class="navbar-brand" href="#">Site</a> <button class="navbar-toggler" data-bs-toggle="collapse"
data-bs-target="#nav"></button> <div class="collapse navbar-collapse" id="nav"> <ul
class="navbar-nav"> <li class="nav-item"><a class="nav-link" href="#home">Home</a></li> <li
class="nav-item"><a class="nav-link" href="#about">About</a></li> <li class="nav-item dropdown">
<a class="nav-link dropdown-toggle" data-bs-toggle="dropdown">Projects</a> <ul
class="dropdown-menu"> <li><a class="dropdown-item" href="#web">Web</a></li> <li><a
class="dropdown-item" href="#ml">ML</a></li> </ul> </li> <li class="nav-item"><a class="nav-link"
href="#contact">Contact</a></li> </ul> </div> </div> </nav>
```

8.b MongoDB Student Queries

```
// insert sample students
db.students.insertMany([{name:'A',gpa:9},{name:'B',gpa:7},{name:'C',gpa:8.5}]); // (i) GPA > 8
db.students.find({gpa:{$gt:8}}); // (ii) Max GPA db.students.aggregate([{ $group: { _id: null,
maxGPA: { $max: "$gpa" } } }]);
```

9.a Library Books (MongoDB)

```
db.books.insertMany([ {title:'Book1', author:'A', genre:'Mystery', year:1995, available:false, copies:2}, {title:'Book2', author:'B', genre:'Thriller', year:2010, available:true, copies:3}, {title:'Book3', author:'C', genre:'Mystery', year:2005, available:true, copies:1}, {title:'Book4', author:'D', genre:'Romance', year:2018, available:true, copies:4}, {title:'Book5', author:'E', genre:'Thriller', year:2016, available:true, copies:2} ]); // (ii) Mystery or Thriller db.books.find({genre:{$in:['Mystery','Thriller']}}); // (iii) Delete before 2000 db.books.deleteMany({year:{$lt:2000}});
```

9.b Products and Contact Servers (Express)

```
// products-server.js (port 5000) const express = require('express'); const app = express(); app.get('/products',(req,res)=>res.json([{id:1,name:'Mobile'}])); app.listen(5000); // contact-server.js (port 3000) const express = require('express'); const app2 = express(); app2.get('/contact',(req,res)=>res.send('Contact: help@shop.com')); app2.listen(3000);
```

10.a Registration with Bootstrap Validation (client-side)

```
<form id="reg" novalidate> <input id="username" required placeholder="Username" /> <div
class="invalid-feedback">Required</div> <input id="email" type="email" required
placeholder="Email" /> <div class="invalid-feedback">Valid email</div> <input id="password"
type="password" minlength="6" required placeholder="Password" /> <div
class="invalid-feedback">Min 6 chars</div> <input id="phone" pattern="\d{10}" placeholder="Phone"
/> <button type="submit">Register</button> </form> <script> const
form=document.getElementById('reg'); form.addEventListener('submit', e=>{
if(!form.checkValidity()){ e.preventDefault(); form.classList.add('was-validated'); } else{
e.preventDefault(); alert('Registered'); } }); </script>
```

10.b Employee Collection (MongoDB)

```
db.employee.insertMany([ {name:'Raj', designation:'Manager', address:'Delhi', salary:55000},
{name:'Simran', designation:'HR', address:'Mumbai', salary:40000}, {name:'Rohit',
designation:'Dev', address:'Chennai', salary:35000}, {name:'Asha', designation:'Tester',
address:'Pune', salary:32000}, {name:'Vikram', designation:'Admin', address:'Kolkata',
salary:28000} ]); // (ii) salary > 30000 db.employee.find({salary:{$gt:30000}}); // (iii) update
addresses db.employee.updateOne({name:'Raj'},{$set:{address:'Hyderabad'}});
db.employee.updateOne({name:'Rohit'},{$set:{address:'Bengaluru'}});
```