Simplified Lab Solutions (With Editable To-Do App)

1.a Bootstrap Responsive Layout (.container vs .container-fluid)

```
<div class="container bg-light p-3 mb-3 border"> <h2>Fixed .container</h2> <p>Content inside
.container has fixed width.</p> </div> <div class="container-fluid bg-info text-white p-3
border"> <h2>Full-width .container-fluid</h2> <p>Content spans full screen width.</p> </div>
```

1.b React To-Do List (Add, Edit, Delete, Complete, LocalStorage)

```
import React, { useState, useEffect } from 'react'; function TodoApp() { const [tasks, setTasks]
= useState(() => JSON.parse(localStorage.getItem('tasks')) || []); const [title, setTitle] =
useState(''); const [editIndex, setEditIndex] = useState(null); useEffect(() => {
localStorage.setItem('tasks', JSON.stringify(tasks)); }, [tasks]); const addTask = () => { if
(!title) return; if (editIndex !== null) { const newTasks = [...tasks]; newTasks[editIndex].title
= title; setTasks(newTasks); setEditIndex(null); } else { setTasks([...tasks, { title, done:
false }]); } setTitle(''); }; const toggleTask = (i) => { const newTasks = [...tasks];
newTasks[i].done = !newTasks[i].done; setTasks(newTasks); }; const deleteTask = (i) =>
setTasks(tasks.filter((_, x) => x !== i)); const editTask = (i) => { setTitle(tasks[i].title);
setEditIndex(i); }; return ( <div style={{ margin: '30px' }}> <h2>■ To-Do List</h2> <input
value={title} onChange={e => setTitle(e.target.value)} placeholder="Enter task..." /> <button
onClick={addTask}>{editIndex !== null ? 'Update' : 'Add'}</button> <ul> {tasks.map((t, i) => (
<li key={i}> <span style={{ textDecoration: t.done ? 'line-through' : 'none' }}>{t.title}</span>
<button onClick={() => toggleTask(i)}>✔</button> <button onClick={() => editTask(i)}>✏</button>
<button onClick={() => deleteTask(i)}>■</button> </li> ))} </ul> </div> ); } export default
TodoApp;
```

2.a Student Details Table (Bootstrap)

```
<table class="table table-striped table-bordered">
<thead><tr><th>Name</th><th>Roll</th><th>Dept</th><th>Email</th></tr></thead>
<tbody><tr><td>Alice</td><td>101</td><td>IT</td><td>a@mail.com</td></tr></tbody> </table>
```

2.b React Controlled Signup Form

```
import React, {useState} from 'react'; function Signup(){ const
[f,setF]=useState({name:'',email:'',pass:''}); const
ch=e=>setF({...f,[e.target.name]:e.target.value}); const sub=e=>{e.preventDefault();alert('Signed
up: '+f.name)}; return(<form onSubmit={sub}> <input name="name" onChange={ch}
placeholder="Name"/><br/> <input name="email" onChange={ch} placeholder="Email"/><br/> <input
name="pass" type="password" onChange={ch} placeholder="Password"/><br/> <button>Submit</button>
</form>); }
```

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">Danger</button>
```

3.b React Auth Form

```
import React,{useState}from'react'; function Auth(){ const[u,setU]=useState({user:'',pass:''});
const ch=e=>setU({...u,[e.target.name]:e.target.value}); return(<form
onSubmit={e=>{e.preventDefault();alert('User: '+u.user)}}> <input name="user" onChange={ch}
placeholder="Username"/><br/> <input name="pass" type="password" onChange={ch}
placeholder="Password"/><br/> <button>Login</button> </form>); }
```

4.a Bootstrap Alert

```
<div class="alert alert-success alert-dismissible fade show"> Form submitted successfully!
<button type="button" class="btn-close" data-bs-dismiss="alert"></button> </div>
```

4.b Employee Payroll MongoDB

```
db.emp.insertOne({name:'Raj',salary:50000}); db.emp.find();
db.emp.updateOne({name:'Raj'},{$set:{salary:55000}}); db.emp.deleteOne({name:'Raj'});
```

5.b Node Calculator Module

```
exports.add=(a,b)=>a+b; exports.sub=(a,b)=>a-b;
```

6.b Node.js Register (GET/POST)

```
const e=require('express');const a=e();a.use(e.urlencoded({extended:true}));
a.get('/',(r,s)=>s.send('<form method=post><input name=name/><button>OK</button></form>'));
a.post('/',(r,s)=>s.send('Welcome '+r.body.name));a.listen(3000);
```

7.a Node HTTP Server

```
const h=require('http'); h.createServer((r,s)=>{ if(r.url==='/courses')s.end('Courses:BCA,BSC');
else if(r.url==='/departments')s.end('Departments:IT,CSE'); else s.end('Welcome');
}).listen(3000);
```

8.b MongoDB GPA

```
db.stu.insertMany([{name:'A',gpa:9},{name:'B',gpa:8.5}]); db.stu.find({gpa:{$gt:8}});
db.stu.aggregate([{$group:{_id:null,maxGPA:{$max:"$gpa"}}}]);
```

9.a Library Books

```
db.books.insertMany([{title:'B1',genre:'Mystery',year:2000},{title:'B2',genre:'Thriller',year:20
15}]); db.books.find({genre:{$in:['Mystery','Thriller']}});
db.books.deleteMany({year:{$lt:2000}});
```

10.a Bootstrap Form Validation

```
<form id="f" novalidate> <input required placeholder="Username"/><div
class="invalid-feedback">Required</div> <button>Submit</button> </form> <script> f.addEventListe
ner('submit',e=>{if(!f.checkValidity()){e.preventDefault();f.classList.add('was-validated');}})
</script>
```