Web Technology Practicals (Code Only)

Practical 1 – Media Gallery using HTML5 Canvas

```
<video id=v width=360 controls src="sample.mp4"></video>
<canvas id=c width=360 height=200></canvas>
<button onclick="(ctx=document.getElementById('c').getContext('2d')).drawImage(document.getElementByI
```

Practical 2 – Dark Mode Toggle

```
<button id=t>Toggle</button>
<script>
const b=document.body;
if(localStorage.dark==='true')b.classList.add('dark');
document.getElementById('t').onclick=()=>{b.classList.toggle('dark');localStorage.dark=b.classList.co
</script>
<style>.dark{background:#111;color:#fff}</style>
```

Practical 3 – Pricing Table using Flexbox

```
<div style="display:flex;gap:10px;flex-wrap:wrap">
<div style="padding:12px;border:1px solid #ccc">Basic<br>■199</div>
<div style="padding:12px;border:1px solid #ccc">Pro<br>■499</div>
<div style="padding:12px;border:1px solid #ccc">Enterprise<br>Contact</div>
</div>
```

Practical 4 – Form Validation using HTML5

```
<form onsubmit="if(!this.checkValidity()){alert('Invalid');return false}">
<input required pattern="\S+@\S+\.\S+" placeholder="email">
<input required minlength=6 placeholder="password">
<button>Submit</button>
</form>
```

Practical 5 – Responsive Layout using Flexbox

```
<style>.wrap{display:flex}.nav{width:180px}.main{flex:1;padding:10px}</style>
<div class="wrap"><nav class="nav">Menu</nav><main class="main"><h1>Dashboard</h1></main></div>
```

Practical 6 – Display Product List using JavaScript

```
<ul id=products></ul>
<script>
const products=[{id:1,name:'Item A'},{id:2,name:'Item B'}];
const ul=document.getElementById('products');
products.forEach(p=>{let li=document.createElement('li');li.textContent=p.name;ul.appendChild(li)});
</script>
```

Practical 7 – Redux Store Setup

```
import { createSlice, configureStore } from '@reduxjs/toolkit';
const cart = createSlice({ name:'cart', initialState:[], reducers:{ add:(s,a)=>{s.push(a.payload)} }
export const { add } = cart.actions;
export const store = configureStore({ reducer:{ cart: cart.reducer }});
```

Practical 8 – Login API Call using Fetch

```
async function login(user,pass){
const r = await fetch('/api/login',{method:'POST',headers:{'Content-Type':'application/json'},body:JS
const j = await r.json();
localStorage.setItem('token', j.token);
}
```

Practical 9 – Multi-Step Form

```
<div id=step1><input id=name required><button onclick="if(name.value){step1.style.display='none';step
<div id=step2 style="display:none"><input id=email required><button onclick="alert('Submitted')">Subm
```

Practical 10 – Laravel API Resource Route

```
Route::middleware('auth:sanctum')->apiResource('products', ProductController::class);
```

Practical 11 – Store Product in Laravel Controller

```
public function store(Request $r){
$r->validate(['name'=>'required']);
return Product::create($r->only('name','price','desc'));
}
```

Practical 12 – MongoDB Aggregation Example

```
db.orders.aggregate([
{ $match: { status: 'complete' } },
{ $group: { _id: '$productId', total: { $sum: '$amount' }, count: { $sum: 1 } } },
{ $sort: { total: -1 } }
]);
```

Practical 13 – Laravel Job Dispatch Example

```
use App\Jobs\ProcessImage;
ProcessImage::dispatch($imagePath);
```

Practical 14 – MongoDB Model in Laravel

```
use Jenssegers\Mongodb\Eloquent\Model as Eloquent;
class Product extends Eloquent {
protected $collection = 'products';
protected $fillable = ['name','price','meta'];
}
```

Practical 15 – Role and Permission Check

```
$user->assignRole('admin');
if(Gate::allows('manage-users')){ /* allowed */ }
```