MERN Stack Revision Kit

```
 Personalized MERN Revision Sheet (Markdown Version)

1.  Project Setup (Backend)
---------------------------------
mkdir backend && cd backend
npm init -y
npm install express mongoose dotenv cors nodemon

Folder Structure:
backend/
   controllers/
   models/
   routes/
   middleware/
   .env
   server.js

server.js Example:
const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');
require('dotenv').config();

const app = express();
app.use(cors());
app.use(express.json());

mongoose.connect(process.env.MONGO_URI)
  .then(() => app.listen(5000, () => console.log('Server running')))
  .catch(err => console.log(err));

2.  React Setup (Frontend)
-------------------------------
npm create vite@latest frontend --template react
cd frontend
npm install react-router-dom axios

Main Routing:
import { BrowserRouter, Routes, Route } from 'react-router-dom';
```

MERN Stack Revision Kit

```
3.  Authentication Flow
-------------------------------
- Hash passwords using bcrypt
- Issue token with jsonwebtoken
- Store token in localStorage
- Use axios.interceptors to attach token

Protected Route:
const token = localStorage.getItem('token');
axios.get('/api/private', { headers: { Authorization: `Bearer ${token}` } });

4.  CRUD Flow Example
-------------------------------
Model:
const mongoose = require('mongoose');
const itemSchema = new mongoose.Schema({ name: String });
module.exports = mongoose.model('Item', itemSchema);

Route:
router.get('/', getItems);
router.post('/', addItem);
router.delete('/:id', deleteItem);

5.  Deployment
-------------------------------
- Use Vercel for frontend
- Use Render for backend
- Store MongoDB URI in environment variables

 MERN Interview Q&A
-------------------------------
1. What is the MERN stack?
   - MongoDB, Express.js, React.js, Node.js

2. How do you secure routes in Express?
   - Use JWTs + middleware to verify token

3. What are React Hooks?
   - useState, useEffect, useContext, useRef, useReducer

4. Difference between useEffect and useLayoutEffect?
```

   - useEffect runs after paint, useLayoutEffect runs before

5. How does React Router work?
   - Manages browser history and renders components based on path

6. MongoDB vs MySQL?
   - Mongo is NoSQL, uses documents, flexible schema

7. Explain middleware in Express.
   - Functions that execute in request/response cycle (e.g., auth, logging)

8. How do you manage state globally in React?
   - Context API or Redux

Flashcards for Code Patterns
----

1. Connecting MongoDB

```
mongoose.connect(process.env.MONGO_URI)
```

2. Protect Route (Express Middleware)

```
const verifyToken = (req, res, next) => {
  const token = req.headers["authorization"]?.split(" ")[1];
  if (!token) return res.status(403);
  jwt.verify(token, process.env.JWT_SECRET, (err, user) => {
    if (err) return res.status(403);
    req.user = user;
    next();
  });
};
```

3. useEffect example

```
useEffect(() => {
  fetchData();
}, []);
```

4. Axios Interceptor

```
axios.interceptors.request.use((config) => {
  config.headers.Authorization = `Bearer ${localStorage.getItem('token')}`;
  return config;
});
```

MERN Stack Revision Kit

```
 Project Checklist
-------------------------------
- [x] MongoDB connection (Atlas)
- [x] Express REST API (CRUD)
- [x] JWT Authentication
- [x] React UI (components, routing)
- [x] Axios API calls
- [x] Protected Routes (frontend + backend)
- [x] Deploy backend (Render/Heroku)
- [x] Deploy frontend (Vercel/Netlify)
- [x] Environment Variables
- [x] Error Handling (try/catch, middleware)
- [x] Form validation
- [x] Search & filter functionality
- [x] Code comments + clean structure
```