**1. Basics of Programming**

* Variables, Data Types
* Loops (for, while), Conditional Statements (if-else, switch)
* Functions (pass by value/reference, recursion)
* Input/Output handling

**2. Complexity Analysis**

* Time Complexity: Big O, Big Theta, Big Omega
* Space Complexity
* Examples: O(1), O(log n), O(n), O(n²)

**3. Core Data Structures (1week)**

**Arrayas**

* Basics: Declaration, Traversal, Insertion, Deletion
* Problems: Prefix Sum, Subarrays, Sliding Window

**Strings**

* Basics: Manipulation, Substrings
* Problems: Palindrome Check, Pattern Matching

**Linked List**

* Singly Linked List: Insert, Delete, Traverse
* Doubly Linked List: Insert, Delete
* Problems: Reverse a Linked List, Detect Cycle

**Stacks**

* Basics: Push, Pop, Peek, Applications
* Problems: Next Greater Element, Valid Parentheses

**Queues**

* Basics: Enqueue, Dequeue, Circular Queue
* Problems: Sliding Window Maximum

**4. Advanced Data Structures (2 weeks)**

**Hashing**

* Hash Tables, Hash Maps
* Problems: Frequency Count, Two Sum

**Trees**

* Binary Trees: Traversals (Inorder, Preorder, Postorder)
* Binary Search Tree (BST): Insert, Search
* Problems: Lowest Common Ancestor, Height of Tree

**Graphs**

* Representation: Adjacency Matrix/List
* Traversals: BFS, DFS
* Problems: Shortest Path, Connected Components

**5. Algorithms (4weeks)**

**Sorting Algorithms**

* Bubble, Selection, Insertion, Merge, Quick Sort
* Applications: Sorting Arrays, Custom Sorting

**Searching Algorithms**

* Linear Search, Binary Search
* Problems: Rotated Array Search

**6. Dynamic Programming (DP)**

* Basics: Recursion + Memoization, Tabulation
* Problems: Fibonacci, Knapsack, Longest Common Subsequence (LCS)

**7. Greedy Algorithms**

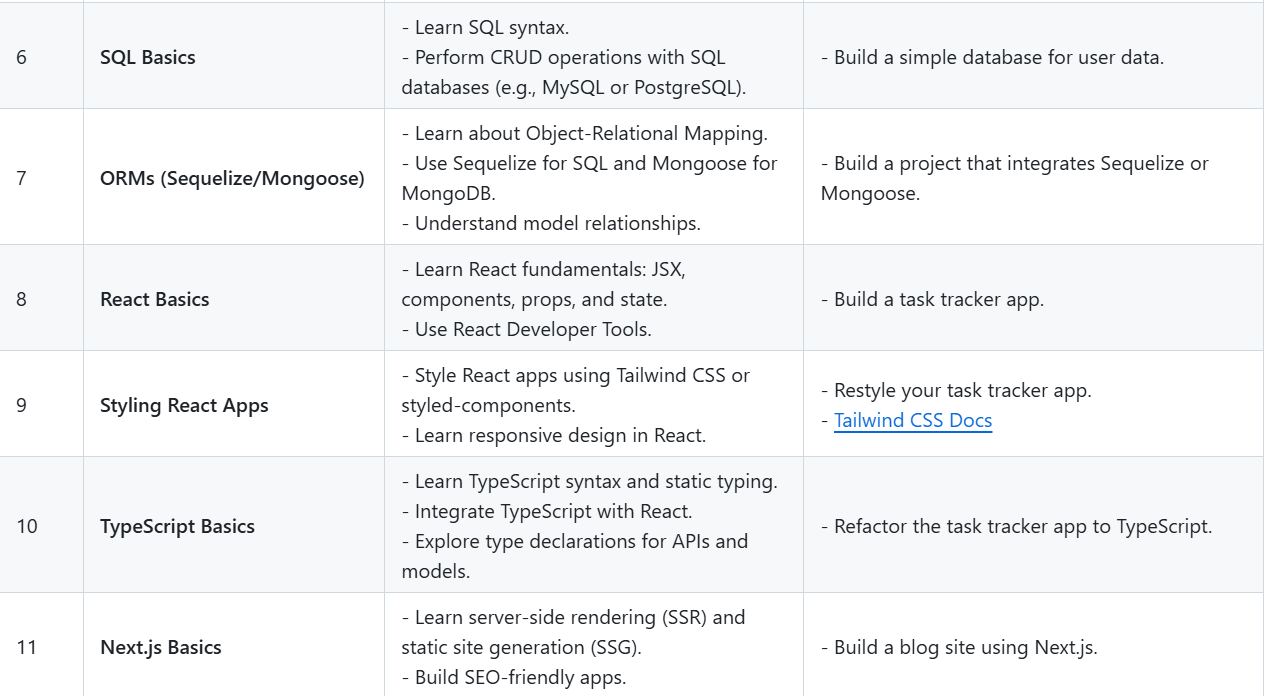
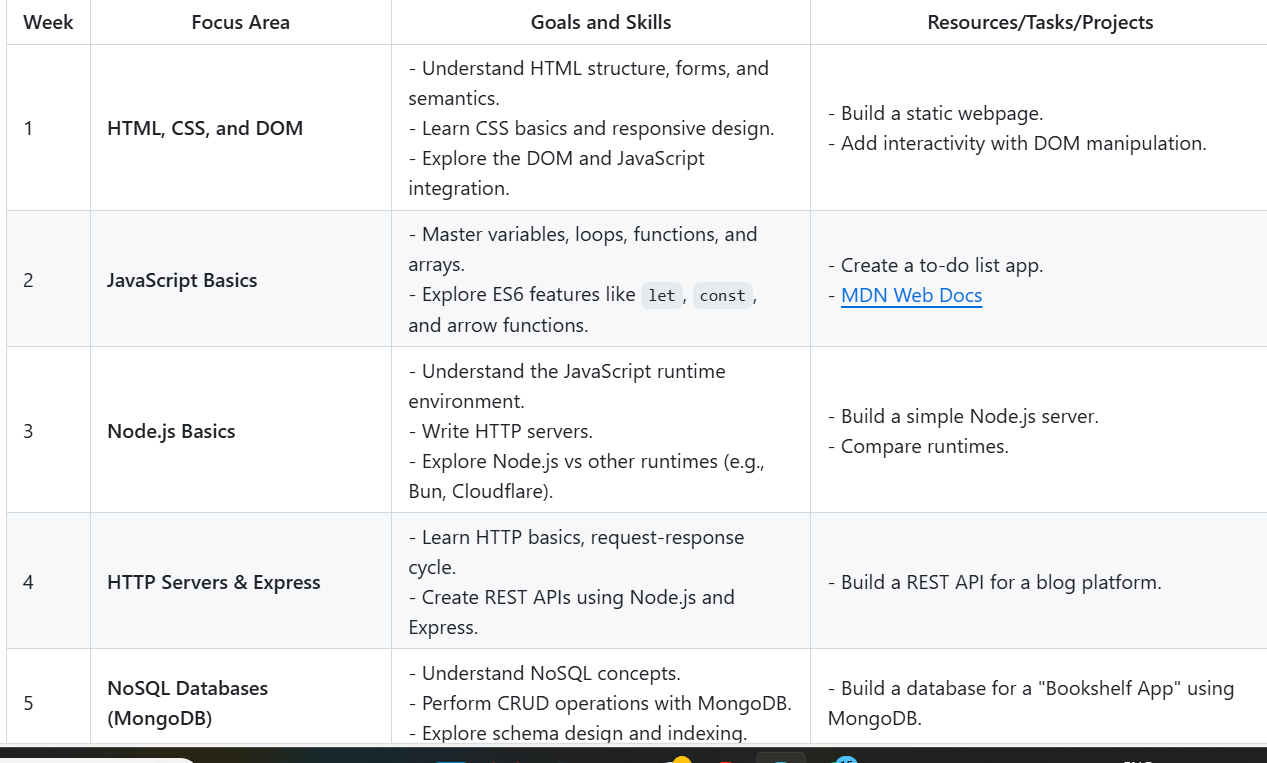
* Interval Scheduling, Huffman Coding
* Problems: Activity Selection, Minimum Spanning Tree (Prim/Kruskal)

**8. Problem-Solving Techniques**

* Two-pointer Technique
* Sliding Window
* Divide and Conquer
* Backtracking (Sudoku Solver, N-Queens)

**9. Advanced Topics (Optional)**

* Tries: Auto-complete, Prefix Matching
* Segment Trees: Range Queries
* Disjoint Sets: Union-Find Algorithm



DEVLOPMENT ROAD MAP