**Project:- CRISPR-Cas9 Target Search**

**Goal:-**

*To simulate how the CRISPR-Cas9 system finds matching DNA sequences using a Trie (prefix tree) data structure.  
CRISPR uses a "guide RNA" to locate specific sequences in the genome for editing. In this project, we replicate this process by storing DNA sequences in a Trie to enable fast and efficient searching for exact or partial matches.*

**Why is this project useful?**

**•** *Explains how computational biology can model gene-editing mechanisms.  
• Helps in understanding CRISPR’s target-finding process.  
• Demonstrates the real-world use of efficient data structures like Trie.  
• Improves understanding of string search optimization in genomics.*

**Skills:-**

*1. Data Structure Design*

*• Implement and use a Trie (prefix tree).  
• Understand how Tries improve search performance for string-based data.*

*2. DNA Sequence Handling*

*• Work with DNA sequences (A, T, G, C).  
• Store, insert, and search biological strings in an optimized format.*

*3. Search Algorithms*

*• Build efficient search for:*

* *Exact match*
* *Prefix/partial match*

*4. Bioinformatics Simulation*

*• Simulate CRISPR-Cas9’s guide RNA matching behavior.  
• Generate target sequence outputs based on user queries.*

**DNA Sequences may use:-**

*• Synthetic DNA sequences from online tools or generated randomly.  
• FASTA format or plain text datasets.  
• (Optional) Real sequences from databases like GenBank or Ensembl.*

**Libraries may use:-**

1. *collections – for TrieNode implementation*
2. *os – to manage files*
3. *(Optional) tkinter or streamlit – for UI (if needed)*

**Expected Outcome**

*• An efficient Trie-based program that can:*

* *Store DNA sequences*
* *Search for CRISPR target matches*
* *Supports exact and partial matching (prefix search).*
* *Lists matched sequences as potential CRISPR cutting targets.*
* *Visual or UI interface for user interaction.*

**Summary:-**

| **Step** | **What You Do** |
| --- | --- |
| **Load DNA Data** | **Accept a list of DNA sequences (strings of A/T/G/C)** |
| **Insert into Trie** | **Store all sequences efficiently using a Trie** |
| **Search Sequences** | **Match exact or partial (prefix) sequences** |
| **Display Targets** | **Show list of matched CRISPR targets** |

**Daily Work Plan (02/07 - 13/07)**

**02/07/2025 – (Project Details Day):**

*• Define the project and finalize objectives  
• Study CRISPR mechanism & Trie data structure  
• Create basic file/folder structure*

**03/07/2025 – (DNA Dataset & Input Format):**

*• Create or collect DNA sequences  
• Understand format: sequence strings like*

*“ATGCGTAC”  
• Store input as .txt or in-memory list  
• Plan functions for inserting and searching*

*sequences*

**04/07/2025 – (Trie Implementation):**

*• Implement TrieNode and Trie classes in Python  
• Test basic insert and search functionalities  
• Add support for exact and prefix matches  
• Design outputs that mimic CRISPR target lists*

**05/07/2025 – (Search Optimization & Matching Logic):**

*• Make prefix search return all matching DNA*

*sequences  
• Ensure accuracy and speed of search logic  
• (Optional) Add case handling for invalid characters*

**06/07/2025 – (Draft 1 Submission):  
Deliverables:**

*• Working Trie insert and search  
• Sample dataset (input and output)  
• Summary of features implemented  
• Code properly organized and commented*

**07/07/2025 – (Advanced Features):**

*• Add sequence validation (only A/T/G/C allowed)  
• Show match count, and exact vs prefix stats  
• Store output matches in a separate .txt or list*

**08/07/2025 – (Visualization / Matching Analysis):**

*• Show which sequences matched a given guide*

*(prefix)  
• (Optional) Compare search time with linear method  
• Add stats like total matches, average match length*

**09/07/2025 – (Report Draft):**

*• Start documentation in Jupyter/Markdown:*

* *Project intro*
* *Trie structure explanation*
* *Sequence handling logic*
* *Example matches*
* *Biological significance of matching*

**10/07/2025 – (Draft 2 Submission):**

**Deliverables:** *• Updated code with advanced logic  
 • Example matching sessions  
 • Clean and structured report  
 • README.md ready*

**11/07/2025 – (Build UI Start):**

*• Create basic command-line or GUI (Tkinter/Streamlit):*

* *Input a DNA query*
* *Display matched targets*
* *Option to add/remove sequences*

**12/07/2025 – (UI Finalization):**

*• Improve layout and labels  
• Add upload file feature (optional)  
• Test UI with multiple input cases  
• Show match count, output clarity*

**13/07/2025 – (Final Project Submission):**

**Deliverables:** *• Final working project folder  
 • Trie class + input handler + search functions  
 • Report or notebook  
 • UI (if created)  
 • Ready for demo or submission*