**Round 2 (R2)-Storyboard**

|  |
| --- |
| **Name of Faculty:** Mr.Praveen Goyal  **Institute:**  Rajkiya Engineering College ,Banda  **Email ID** (as submitted in the registration form) **:** praveen\_soit@yahoo.com  **Discipline to which the Lab belongs: Information Technology**  **Name of the Lab: Data Structure**  **Name of experiment: Tower of Hanoi**  **Kindly Refer these documents before filling the worksheet**  **1. Coursework (MOOC ) on Pedagogy , Storyboard , Lab Manual :**  [**h ttp://bit.ly/Vlabs-MOOC**](http://bit.ly/Vlabs-MOOC) **2. Additional Documentation booklet for reference.** [**http://vlabs.iitb.ac.in/vlabs-dev/document.php**](http://vlabs.iitb.ac.in/vlabs-dev/document.php)  **3. Sample Git Repository. : https://github.com/BootTeam11/Boot2k19.git** |

**Round 2**

**1. Story Outline:**

To understand and code the problem of Tower of Hanoi using recursion for n disks.

1. **Story:**

The Tower of Hanoi is a mathematical game or puzzle. It consists of three rods, and a number of disks of different sizes which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape. The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

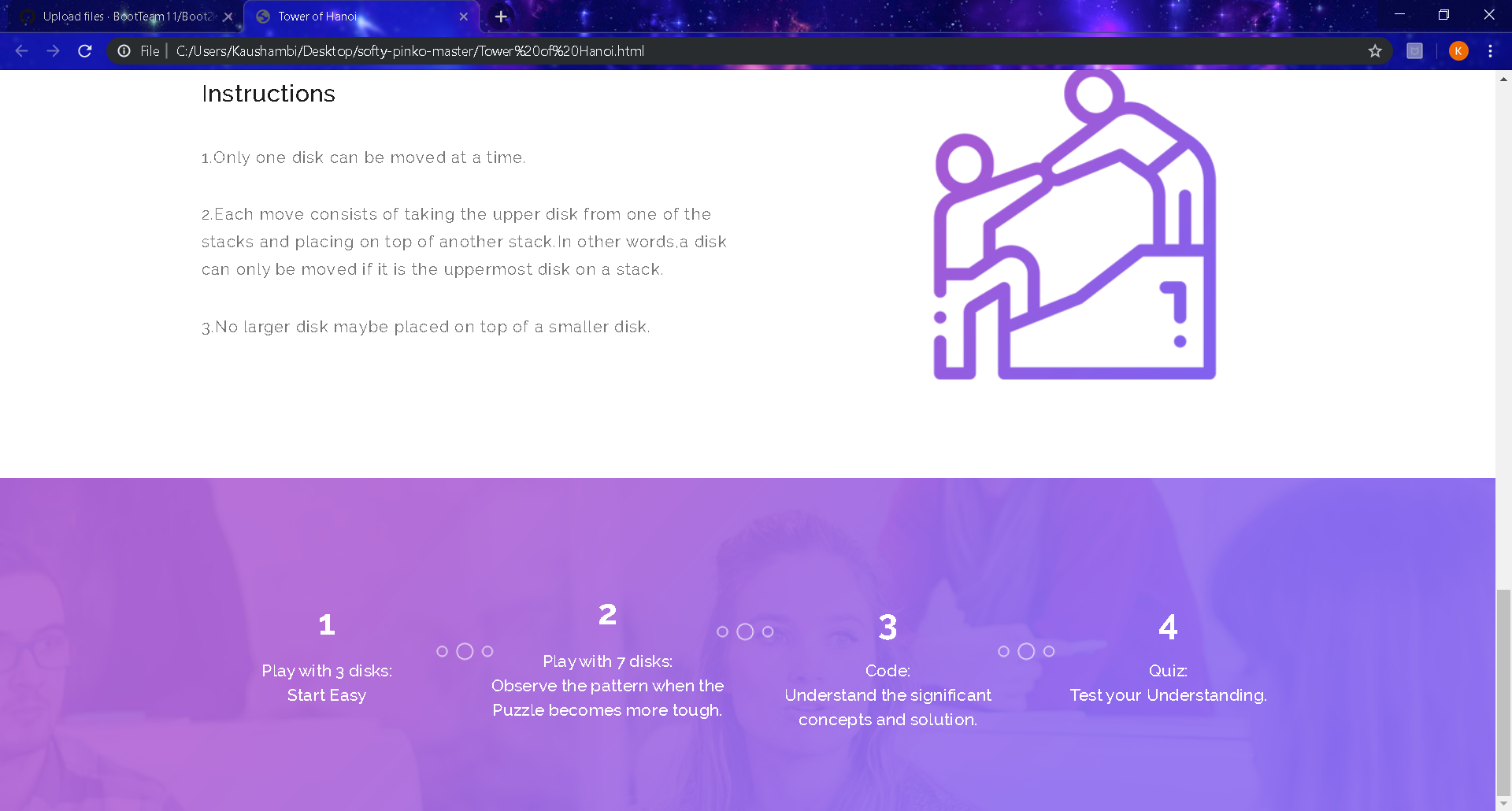
• Only one disk can be moved at a time.

• Each move consists of taking the upper disk from one of the stacks and placing it on top of nother stack i.e. a disk can only be moved if it is the uppermost disk on a stack.

• No disk may be placed on top of a smaller disk. With three disks, the puzzle can be solved in seven moves.

The minimum number of moves required to solve a Tower of Hanoi puzzle is 2 n - 1, where n is the number of disks.

* 1. **Set the Visual Stage Description:**



* 1. **Set User Objectives & Goals:**

* **A Glimpse at the Puzzle**

The student is made aware of the importance of the puzzle through mentioned history.

* **Game Play of 3 disks**

The student is supposed to solve the puzzle consisting of three disks following the instructions.

* **Game Play of 7 disks**

The student is supposed to solve the puzzle consisting of seven disks following the instructions.

* **Coding the Solution**

The student is supposed to understand the approach to solution of the puzzle ad comprehend the code.

* **Quiz**

The student is supposed to answer the questions after playing the games and comprehending the code.

* 1. **Set the Pathway Activities:**

* **Game Play of 3 disks**

The student is supposed to solve the puzzle consisting of three disks following the instructions.

* **Game Play of 7 disks**

The student is supposed to solve the puzzle consisting of seven disks following the instructions.

* **Coding the Solution**

The student is supposed to understand the approach to solution of the puzzle ad comprehend the code.

* 1. **Set Challenges and Questions/Complexity/Variations i n Questions:**

Quiz/Comprehension check: A sequence of questions followed based on gameplay and solution to the code.(Concept of Recursion)

* 1. **Allow pitfalls:**

* + Student is allowed to skip to the solution page in case she/he fails to solve the Puzzle.

**2.6 Conclusion:**

Understand the concept of recursion and use it for solving the puzzle as its application.

**2.7 Equations/formulas:**

* The minimum number of moves required to solve a Tower of Hanoi puzzle is 2^n - 1, where n is the number of disks.
* using namespace std;

void towerOfHanoi(int n, char from\_rod,

                    char to\_rod, char aux\_rod)

{

    if (n == 1)

    {

        cout << "Move disk 1 from rod " << from\_rod <<

                            " to rod " << to\_rod<<endl;

        return;

    }

    towerOfHanoi(n - 1, from\_rod, aux\_rod, to\_rod);

    cout << "Move disk " << n << " from rod " << from\_rod <<

                                " to rod " << to\_rod << endl;

    towerOfHanoi(n - 1, aux\_rod, to\_rod, from\_rod);

}

// Driver code

int main()

{

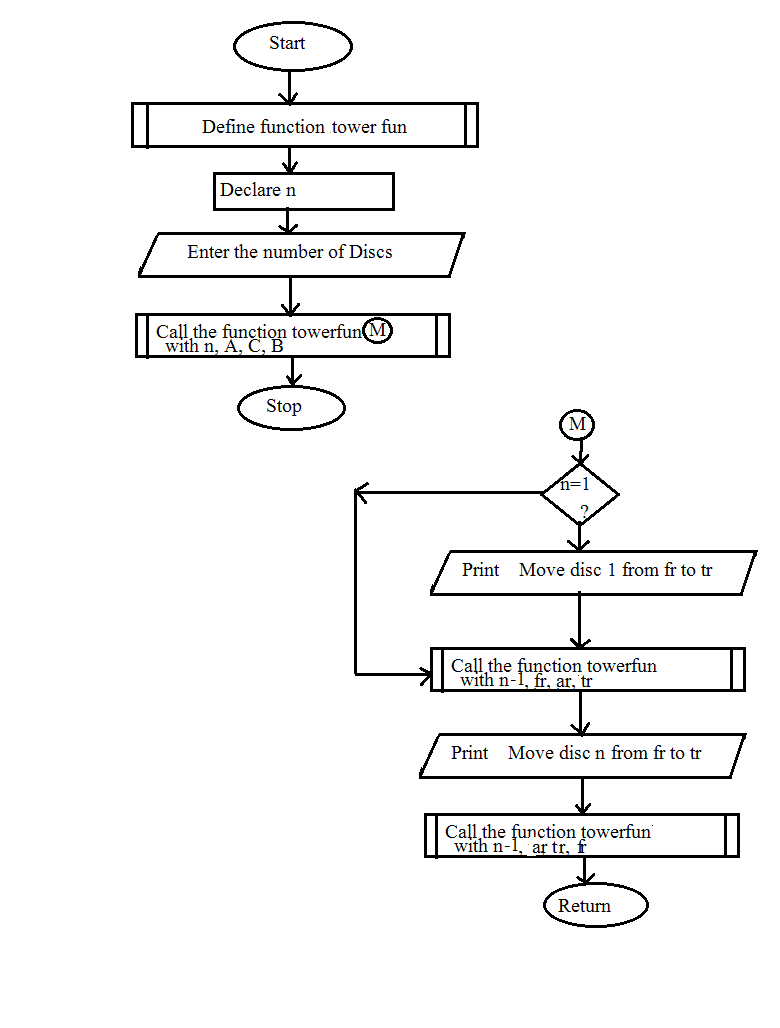
    int n = 4; // Number of disks

    towerOfHanoi(n, 'A', 'C', 'B'); // A, B and C are names of rods

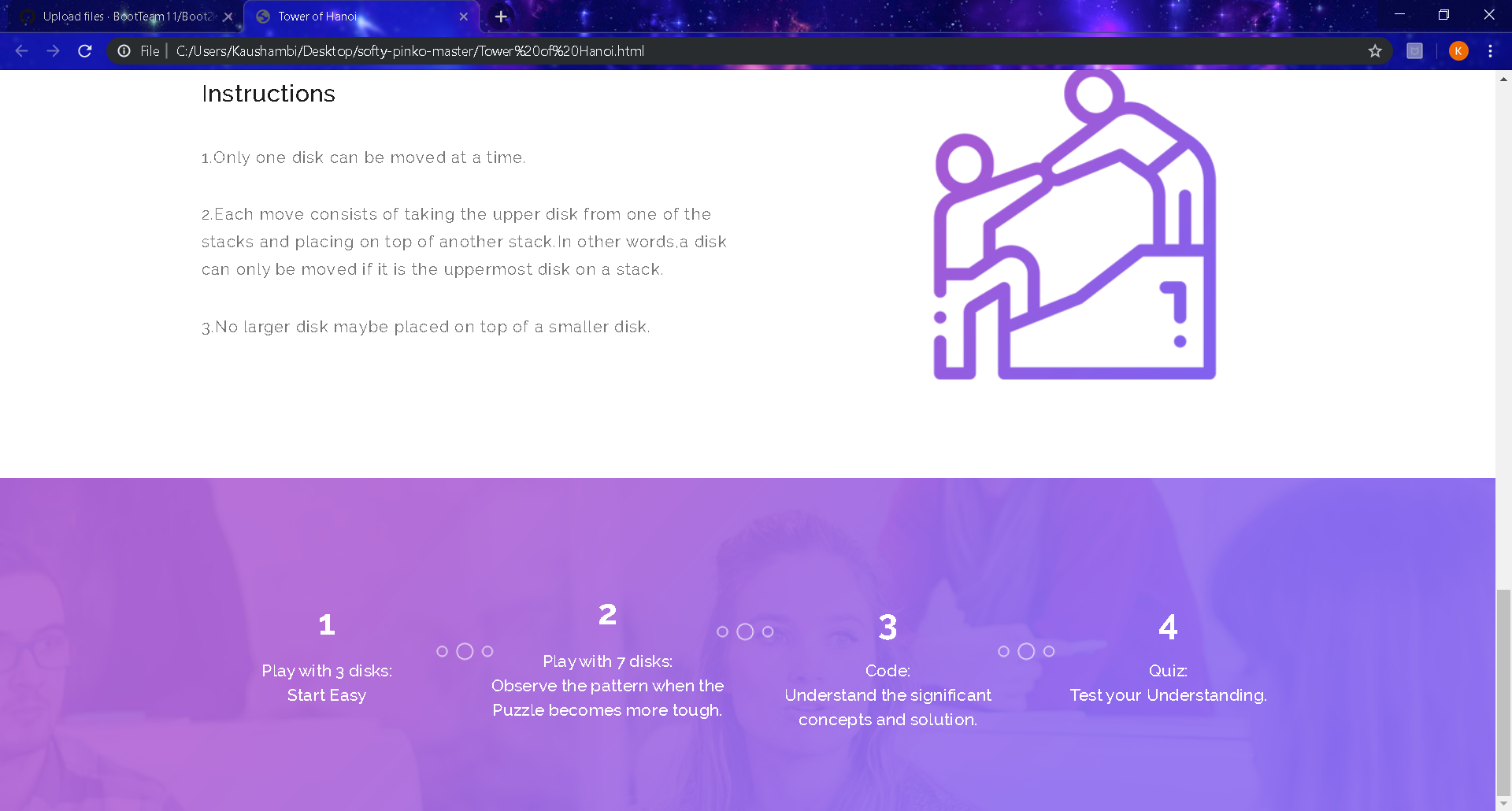
    return 0;

}

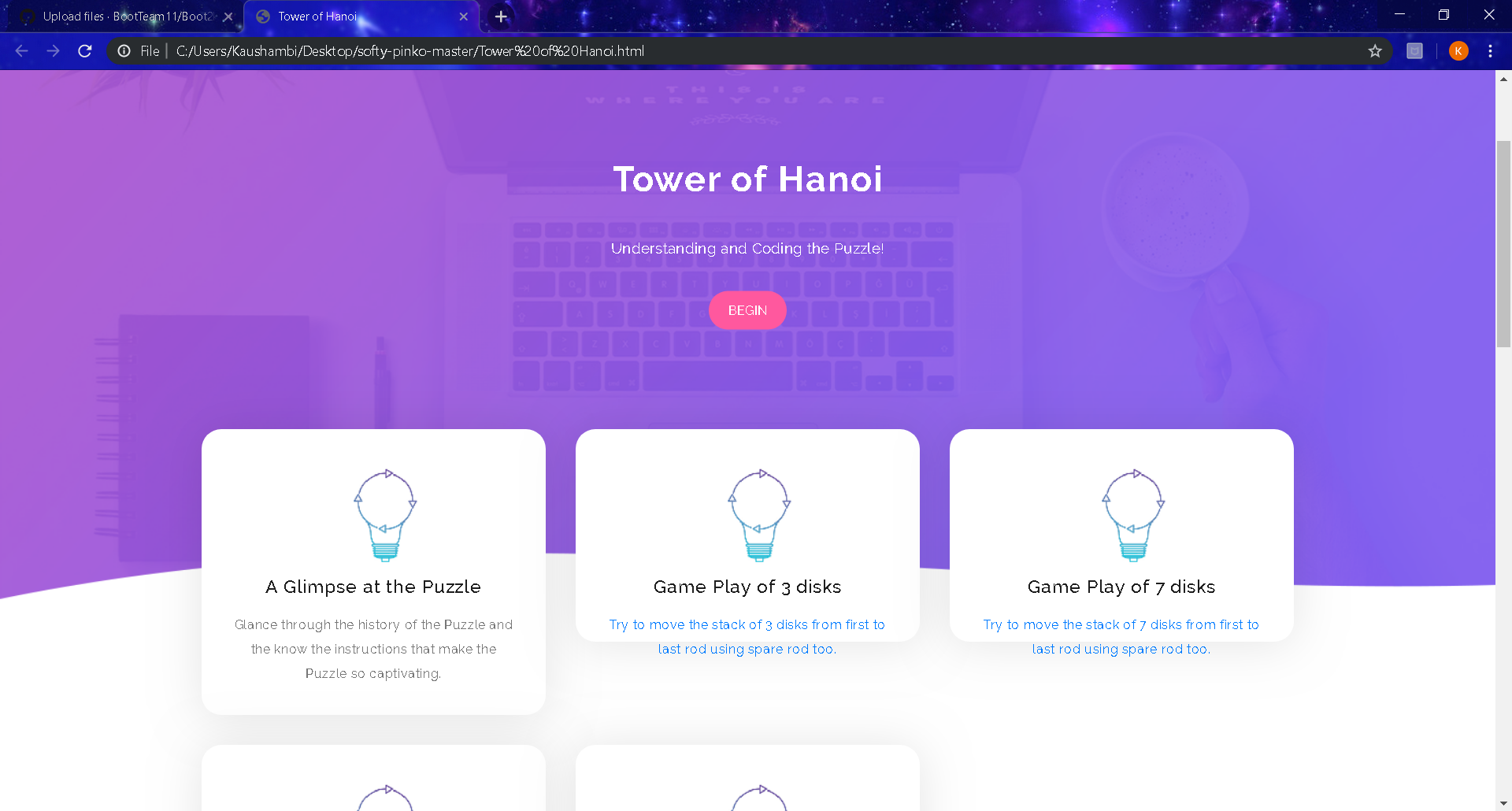
.**Flowchart**

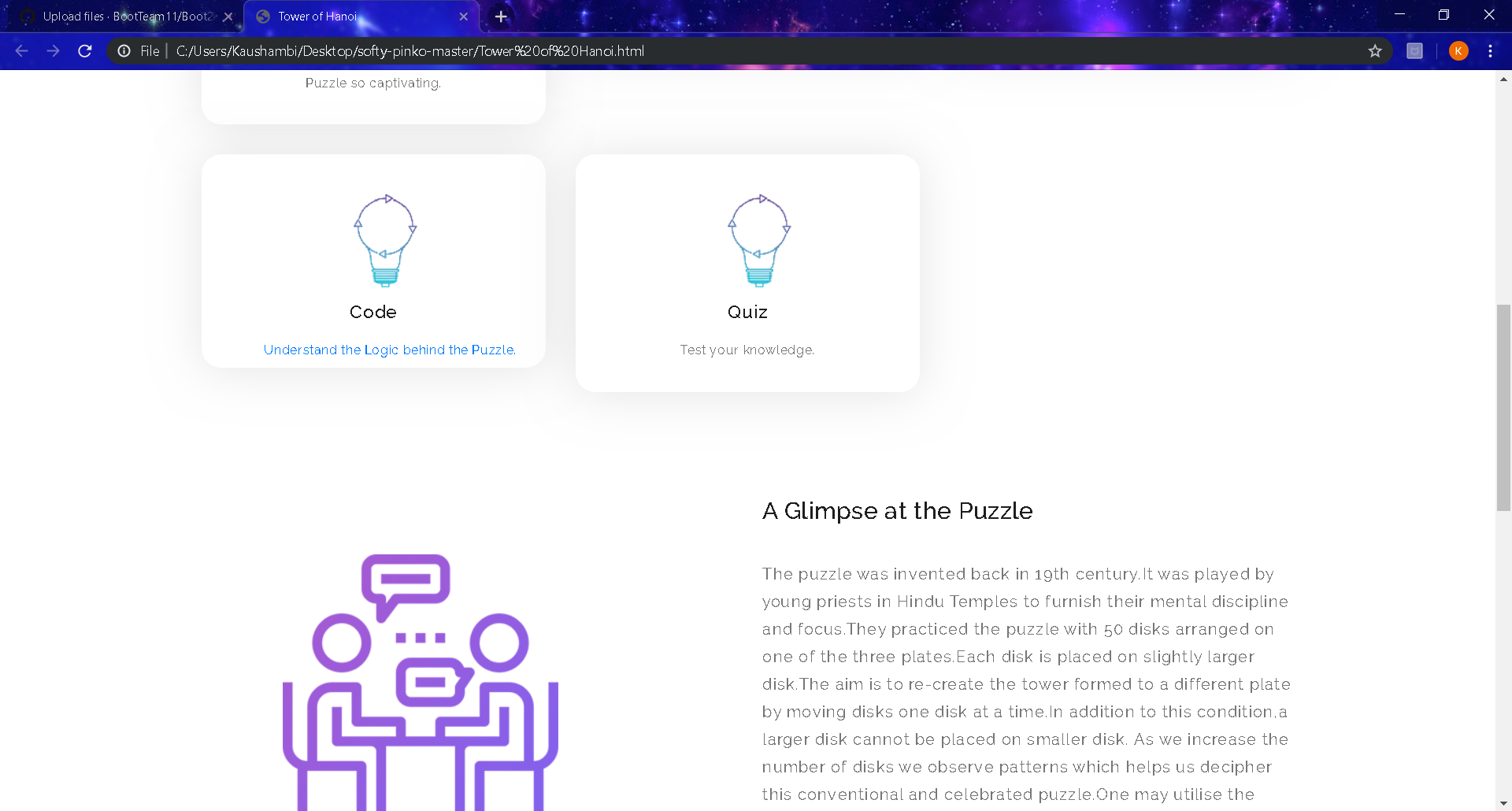


**Mindmap**



1. **Storyboard**





Copyright 2018-19, Virtual Labs - IIT Bombay.