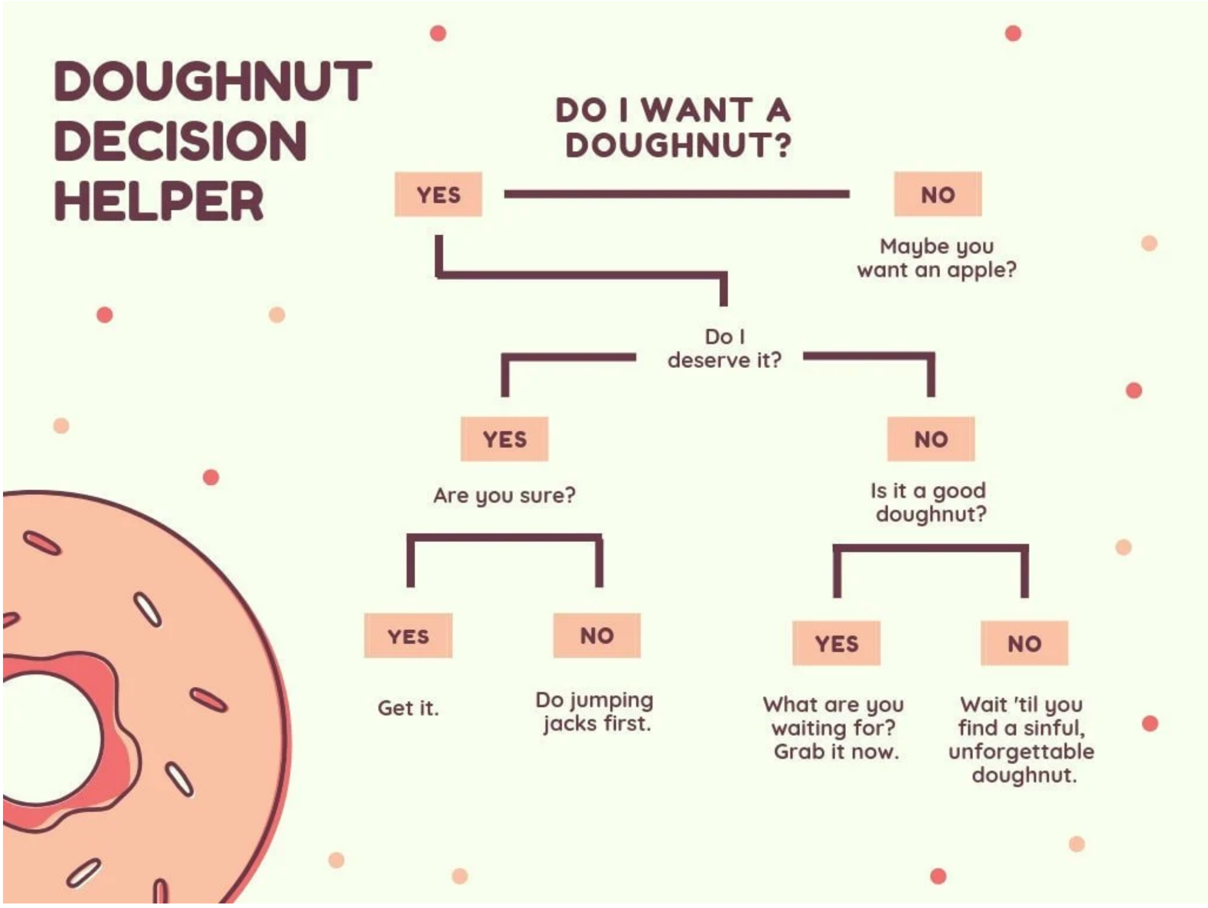
Choose your own adventure

Problem Statement

Write a simple web application which allows a player to choose their own path by picking between two choices displayed on their screen in order to progress to the next set of choices, until they get to one of the endings. You should be able to persist the player's choices and show the steps they took to get to the end of the game.
A major bonus would be a tree diagram page that shows all the possible choices, and highlights the choices the user has made for the story. (These are just some random examples, it doesn't need to look as pretty or as complicated as this.)



This is an open-ended exercise, but we would prefer it if you used .net Core as the back end
and Angular or React as the front end.

- You can choose a cool topic that you're interested in.
- You can make your answers in a Yes/No format or use full sentences.

• In the end, make it fun, for yourself and for the people using it.

My Example Decision Heirarchy

Click here to learn about social networks

1. Twitter
    a. What is twitter?
        i. Twitter is an American microblogging and social networking service on which users post and interact with messages known as "tweets". Click to know more.
            1. Go to google.com and search twitter :)
    b. How to post on twitter?
        i. Go to twitter.com, sign up/sign in and type in some message and click the Send icon. Click here what more you can do on twitter.
            1. Follow other profiles
            2. Chat with the people who follow you
2. Facebook
    a. What is facebook?
        i. Facebook, Inc., is an American multinational technology company based in Menlo Park, California.
    b. How to post on facebook?
        i. Go to facebook.com, sign up/sign in and then on your timeline type in some message and click the Post button

Approach

Let’s discuss the tech stack and a possible architecture to solve this problem.

Technology Stack

Backend

Database: SQL Server
API: .NET Core Web API

Frontend

Vue.js, HTML, CSS, JavaScript

Back End

Database

Name: `Decisions`
Schema: Here we are using one Primary Key `DecisionId`, a `Description` of the decision and a Foreign Key to reference the same table with column `ParentDecisionId` so that we can store child decisions.

Here is the create script.

```
CREATE TABLE Decisions
(
      DecisionId INT PRIMARY KEY IDENTITY(1, 1),
      Description VARCHAR(200) NOT NULL,
      ParentDecisionId INT REFERENCES Decisions(DecisionId)
)
```

Stored Procuedure:
A stored procedure is needed in order to get all decisions with their details ordered by a depth column set as 0 for the root question and then incremented as we load the children.

```
USE [ChooseYourOwnAdventure]
GO

CREATE Procedure GetDecisions
AS

      With DecisionCTE (DecisionId, ParentDecisionId, Description,
Depth)
      As
      (
            Select
                  DecisionId,
                  ParentDecisionId,
                  Description,
                  0 As Depth
            From
                  dbo.Decisions
            Where
                  DecisionId = 1
            Union All

            Select
                  Child.DecisionId,
                  Child.ParentDecisionId,
                  Child.Description,
                  Parent.Depth + 1 As Depth
            From
                  dbo.Decisions As Child
            Join
                  DecisionCTE As Parent
            On
                  Child.ParentDecisionId = Parent.DecisionId
      )

      Select DecisionId, ParentDecisionId, Description, Depth
      From DecisionCTE
      Order By Depth, DecisionId
GO
```

Table

Decisions

Column	**Details**
DecisionId	INT PRIMARY KEY IDENTITY(1, 1)
Description	VARCHAR (200), NOT NULL
ParentDecisionId	INT NULL REFERENCES Decisions (DecisionId)

API

Endpoint

```
GET api/DecisionHelper

Response:
    Content-Type: application/json
```

Response Example

```
{
       "decisionId": 1,
       "description": "Click here to learn about social networks",
       "parentDecisionId": null,
       "depth": 0,
       "parentDecision": null,
       "childDecisions": [{
             "decisionId": 2,
             "description": "Twitter",
             "parentDecisionId": 1,
             "depth": 1,
             "childDecisions": [{
                   "decisionId": 4,
                   "description": "What is twitter?",
                   "parentDecisionId": 2,
                   "depth": 2,
                   "childDecisions": [{
                         "decisionId": 6,
                         "description": "Twitter is an American
microblogging and social networking service on which users post and
interact with messages known as \"tweets\". Click to know more.",
                         "parentDecisionId": 4,
                         "depth": 3,
                         "childDecisions": [{
                               "decisionId": 12,
                               "description": "Go to google.com and search
twitter :)",
                               "parentDecisionId": 6,
                               "depth": 4,
                               "childDecisions": []
                         }]
                   }]
             }, {
                   "decisionId": 5,
                   "description": "How to post on twitter?",
                   "parentDecisionId": 2,
                   "depth": 2,
                   "childDecisions": [{
                         "decisionId": 7,
                         "description": "Go to twitter.com, sign up/sign
in and type in some message and click the Send icon. Click here what
more you can do on twitter.",
                         "parentDecisionId": 5,
                         "depth": 3,
                         "childDecisions": [{
                               "decisionId": 13,
                               "description": "Follow other profiles",
                    "parentDecisionId": 7,
                    "depth": 4,
                    "childDecisions": []
                }, {
                    "decisionId": 14,
                    "description": "Chat with the people who
follow you",
                    "parentDecisionId": 7,
                    "depth": 4,
                    "childDecisions": []
                }]
            }]
        }]
    }, {
        "decisionId": 3,
        "description": "Facebook",
        "parentDecisionId": 1,
        "depth": 1,
        "childDecisions": [{
            "decisionId": 8,
            "description": "What is facebook?",
            "parentDecisionId": 3,
            "depth": 2,
            "childDecisions": [{
                "decisionId": 10,
                "description": "Facebook, Inc., is an American
multinational technology company based in Menlo Park, California.",
                "parentDecisionId": 8,
                "depth": 3,
                "childDecisions": []
            }]
        }, {
            "decisionId": 9,
            "description": "How to post on facebook?",
            "parentDecisionId": 3,
            "depth": 2,
            "childDecisions": [{
                "decisionId": 11,
                "description": "Go to facebook.com, sign up/sign
in and then on your timeline type in some message and click the Post
button.",
                "parentDecisionId": 9,
                "depth": 3,
                "childDecisions": []
            }]
        }]
```

```
      }]
}
```

Front End

The step-by-step logic can be interpreted as below.

1. Call the API endpoint `api/DecisionHelper` and get the of decision with its child decisions.
2. Display the first decision only
3. Render the decision with the description
    a. If there are child decisions, then render the child decisions
        i. With on click of any of these children, display the child decision
        ii. For tracking, emit an event on the bus and send the selected decision id
    b. If no answers are available, then it is a terminal decision
        i. Emit an event on the bus to indicate to render the whole tree
        ii. Render the whole decision tree structure
        iii. Indicate the decisions that are tracked on the tree