A Project Report

on

UNDERSTANDING QUANTUM COMPUTING CONCEPTS BY ACTIVITIES USING WEB APPLICATION

Submitted in partial fulfillment of the

requirement for the award of the degree of

B. Tech C.S.E

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Name of Supervisor: Dr. V. Jayakumar
Designation: Assistant Professor

Submitted By

Siddharth Raj
18SCSE1010014

Prerna Sharma
18SCSE1010625

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING /
DEPARTMENT OF COMPUTERAPPLICATION
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
DECEMBER, 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled **"UNDERSTANDING QUANTUM COMPUTING CONCEPTS BY ACTIVITIES USING WEB APPLICATION"** in partial fulfillment of the requirements for the award of the Bachelor of Technology in Computer Science and Engineering submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of July,2021 to December and 2021, under the supervision of Dr V. Jayakumar (Assistant Professor), Department of Computer Science and Engineering, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Siddharth Raj, 18SCSE1010014
Prerna Sharma, 18SCSE1010625

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor Name

Designation

CERTIFICATE

The Final Project Viva-Voce examination of Siddharth raj 18SCSE1010014 and Prerna Sharma,18SCSE1010625 has been held on 20-12-2021 and his/her work is recommended for the award of Bachelor of Technology in Computer Science and Engineering with Specialization in Internet of Things and Bachelor of Technology in Computer Science and Engineering respectively

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December, 2021

Place: Greater Noida

Abstract

The sole and main idea behind this project is that Quantum Computing is emerging field and people are still struggling to find proper resources to learn quantum computing concepts and the activities-based learning is really very limited. Taking an example if someone wants to start learning or understanding quantum computing and how it works what are the concepts, they have very limited options either they can go for books or they can refer to some online documentation activity-based learning in this field is still missing or very minimal. There are some of resources that are available but all are scattered you can't find anything in just one place you have to go through many of the websites and books that are available to understand these concepts. We decided to implement this Idea in form of website so that any user can easily understand some working concepts of quantum computing all at one place with proper documentation and activities and sorted in a proper manner. For website we use HTML and, CSS for designing and ReactJS as framework, NodeJS for backend and MongoDB for database. This website will make their life easier who wants to learn concepts of quantum computing by providing all resources at one place and they don't need to spend time on searching resources.

Keywords- quantum computing, HTML, CSS, ReactJS, NodeJS, MongoDB.

Table of Contents

List of Table

List of Figures

Acronyms

B.Tech.	Bachelor of Technology
M.Tech.	Master of Technology
BCA	Bachelor of Computer Applications
MCA	Master of Computer Applications
B.Sc. (CS)	Bachelor of Science in Computer Science
M.Sc. (CS)	Master of Science in Computer Science
SCSE	School of Computing Science and Engineering
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
JS	JavaScript
ER	Entity Relationship
UML	Unified Modeling Language

CHAPTER-1

Introduction

Keeping in mind the emerging field of quantum computing and contribution and interest of people in that the demand of resources to learn quantum computing is increasing day by day. Generally, People prefer learn by activities and in many of research and test it is found that learning by activities is very effective. But the resources to learn quantum computing concepts by activities is very limited and to fulfill the demand and keep the enthusiasm of learner alive we came with a solution to make a dynamic website using HTML, CSS, Bootstrap and ReactJS which will help in understanding the Quantum Computing concepts with the help of some interactive activities. There are some resources which are available on internet and also in books but they all are scattered anyone who want to learn something they have search things here and there and then they will get proper content with some activities and explanation and between all these there are a lot of waste of time. We will make all things available on a single website having information, resources, and interactive activities which will help in learning and understanding quantum computing concepts in easy way. This will eventually make life easier and enhance learning enthusiasm of learners.

Technology Used:

HTML: Hypertext Markup Language, the basic function is creating web pages. The goal of the web browser is to read the documents as web-pages; and it is also possible to include scripts written in several languages, such as JavaScript, which an impact on the behavior of web pages [1]

CSS: Cascading Style Sheet, is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. Using this we have decorate this web-pages with awesome colors tables and bars.[2]

JavaScript: A programming language developed for the design of interactive sites and creating web applications. JavaScript can interact effectively with HTML source code, enabling web authors access to their sites with dynamic content.[3]

ReactJS: (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies. React can be used as a base in the development of single page or mobile applications. However, react is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.[4]

Bootstrap: is a free and open-source CSS framework directed at responsive, mobile-first and frontend web development. It contains CSS- and (optionally) JavaScript based design templates for typography, forms, buttons, navigation, and other interface components. Bootstrap is the seventh-most starred project on GitHub, with more than 142,000 stars, behind free-code-camp (almost 312,000 stars) and marginally behind Vue.js framework. Programs used to implement ingredients to recipe are as follows [5]

Visual Studio Code (VS code): is a free source code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.[6]

Google Chrome: Google Chrome is a cross-platform web browser developed by Google. It was first released in 2008 for Microsoft Windows, and was later ported to Linux, macOS, iOS, and Android where it is the default browser built into the OS. The browser is also the main component of Chrome OS, where it serves as the platform for web applications.[7]

CHAPTER 2

LITERATURE SURVEY

This section explores literature review of this system and discuss some of similar available systems. There are some resources available on Internet and also in form of books some of them are:

2.1. Quantum Atlas:

It is a website for studying quantum technologies which also addresses some concepts of quantum computing, Qubits, Superposition the prestation and explanations available are very good they're explaining things using some beautiful illustrations you can visit the website on https://quantumatlas.umd.edu/.

Quantum physics governs everything that's very small or very cold, and it has a reputation for being extraordinarily difficult to understand. We want to change that.

With The Quantum Atlas, we aim to provide an approachable guide to quantum physics for non-experts—anyone looking for an explanation of a quantum concept they read about in an article or heard about on the radio. To that end, we've created images, animations, short podcasts and written explanations of the most fundamental quantum concepts, and we're planning to add more as The Atlas grows.

The Quantum Atlas is a place where anyone can learn about quantum physics. Check out the Get Started page to get started!

The Quantum Atlas is supported by a grant from the National Science Foundation's Advancing Informal STEM Learning program (Award #1713387). It is also supported by the Joint Quantum Institute, a research partnership between the University of Maryland and the National Institute of Standards and Technology with the support and participation of the Laboratory for Physical Sciences, and the Illinois Quantum Information Science and Technology Center.[8]

2.2. Books

There are plenty of books available which elaborate the concepts using images like "Quantum Computing for the Quantum Curious" [9] by Ciaran Hughes, Joshua Isaacson, Anastasia Perry, Ranbel F. Sun, Jessica

Turner. But the main issue that arises that these are not that interactive you'll get the theoretical knowledge but for actual interaction you have to go some other resources

This open access book makes quantum computing more accessible than ever before. A fast-growing field at the intersection of physics and computer science, quantum computing promises to have revolutionary capabilities far surpassing "classical" computation. Getting a grip on the science behind the hype can be tough: at its heart lies quantum mechanics, whose enigmatic concepts can be imposing for the novice.

This classroom-tested textbook uses simple language, minimal math, and plenty of examples to explain the three key principles behind quantum computers: superposition, quantum measurement, and entanglement. It then goes on to explain how this quantum world opens up a whole new paradigm of computing.

The book bridges the gap between popular science articles and advanced textbooks by making key ideas accessible with just high school physics as a prerequisite. Each unit is broken down into sections labelled by difficulty level, allowing the course to be tailored to the student's experience of math and abstract reasoning. Problem sets and simulation-based labs of various levels reinforce the concepts described in the text and give the reader hands-on experience running quantum programs.

This book can thus be used at the high school level after the AP or IB exams, in an extracurricular club, or as an independent project resource to give students a taste of what quantum computing is really about. At the college level, it can be used as a supplementary text to enhance a variety of courses in science and computing, or as a self-study guide for students who want to get ahead. Additionally, readers in business, finance, or industry will find it a quick and useful primer on the science behind computing's future.

CHAPTER-3

PROPOSED WORK

This section explores the system we're proposing to make users life easy. The most important steps that have been taking to build the web application for learning quantum computing with the help of activities.

Our website will provide following features:

- User can learn the concepts by live illustrations. People prefer learn by activities and in many of research and test it is found that learning by activities is very effective. But the resources to learn quantum computing concepts by activities is very limited and to fulfill the demand and keep the enthusiasm of learner alive we came with a solution to make a dynamic website using HTML, CSS, Bootstrap and ReactJS which will help in understanding the Quantum Computing concepts with the help of some interactive activities. There are some resources which are available on internet and also in books but they all are scattered anyone who want to learn something they have search things here and there and then they will get proper content with some activities and explanation and between all these there are a lot of waste of time. We will make all things available on a single website having information, resources, and interactive activities which will help in learning and understanding quantum computing concepts in easy way. This will eventually make life easier and enhance learning enthusiasm of learners.
- User can understand some of the concepts by playing games or activities:
  There are some activities that are included in the system to understand the concept in a better way like for example quantum tic tac toe: Quantum tic-tac-toe is a "quantum generalization" of tic-tac-toe in which the players' moves are "superpositions" of plays in the classical game. The game was invented by Allan Goff of Novatia Labs, who describes it as "a way of introducing quantum physics without mathematics", and offering "a conceptual foundation for understanding the meaning of quantum mechanics in this game instead of having single move at a time for each player like traditional tic-tac-toe game, in quantum tic-tac-toe each player has two moves which describes the concept superposition we will discuss this further in detail in this report.

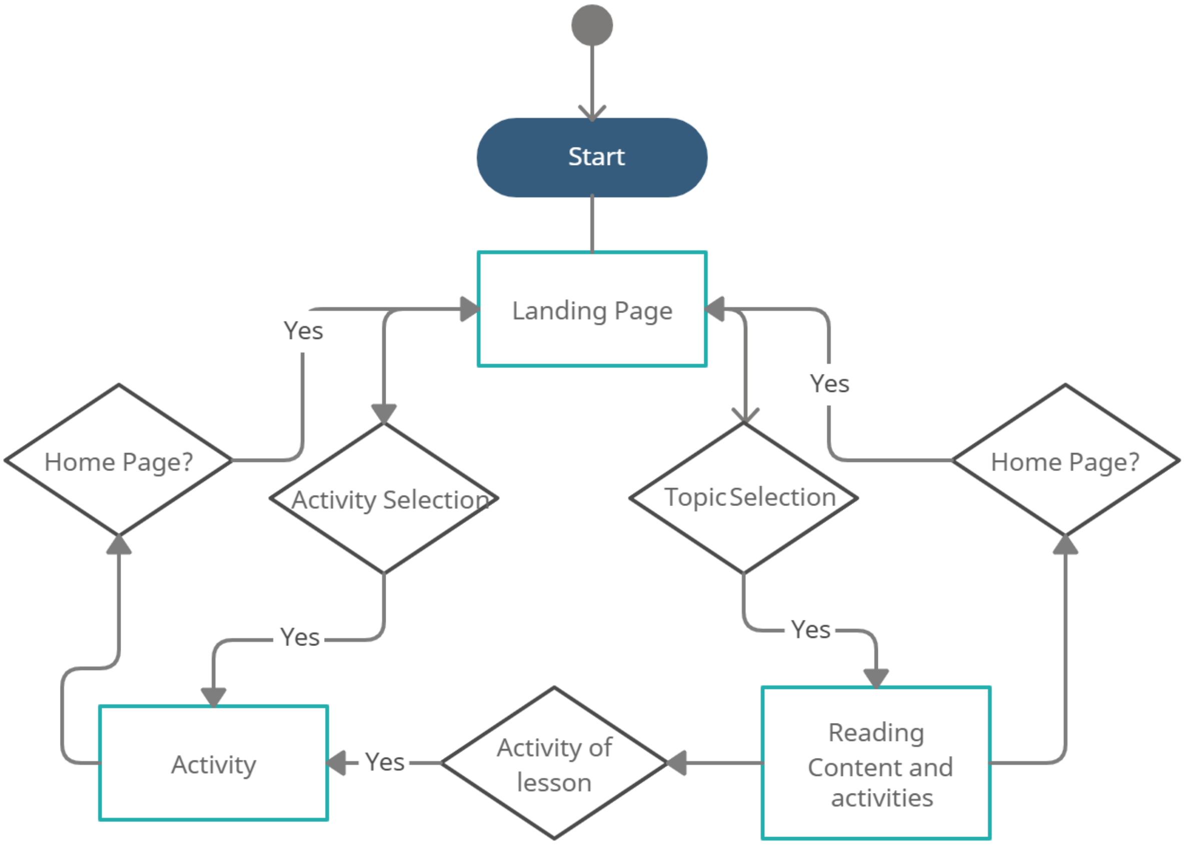


Fig 1: Architectural Diagram of System

This diagram demonstrates and architectural over view of the Proposed System in This website will have only one actor i.e., user and it will require no authentication at this stage. User: can see the dashboard in which they can easily select the lesson and activity they can also give feedback and report any bug to developer.

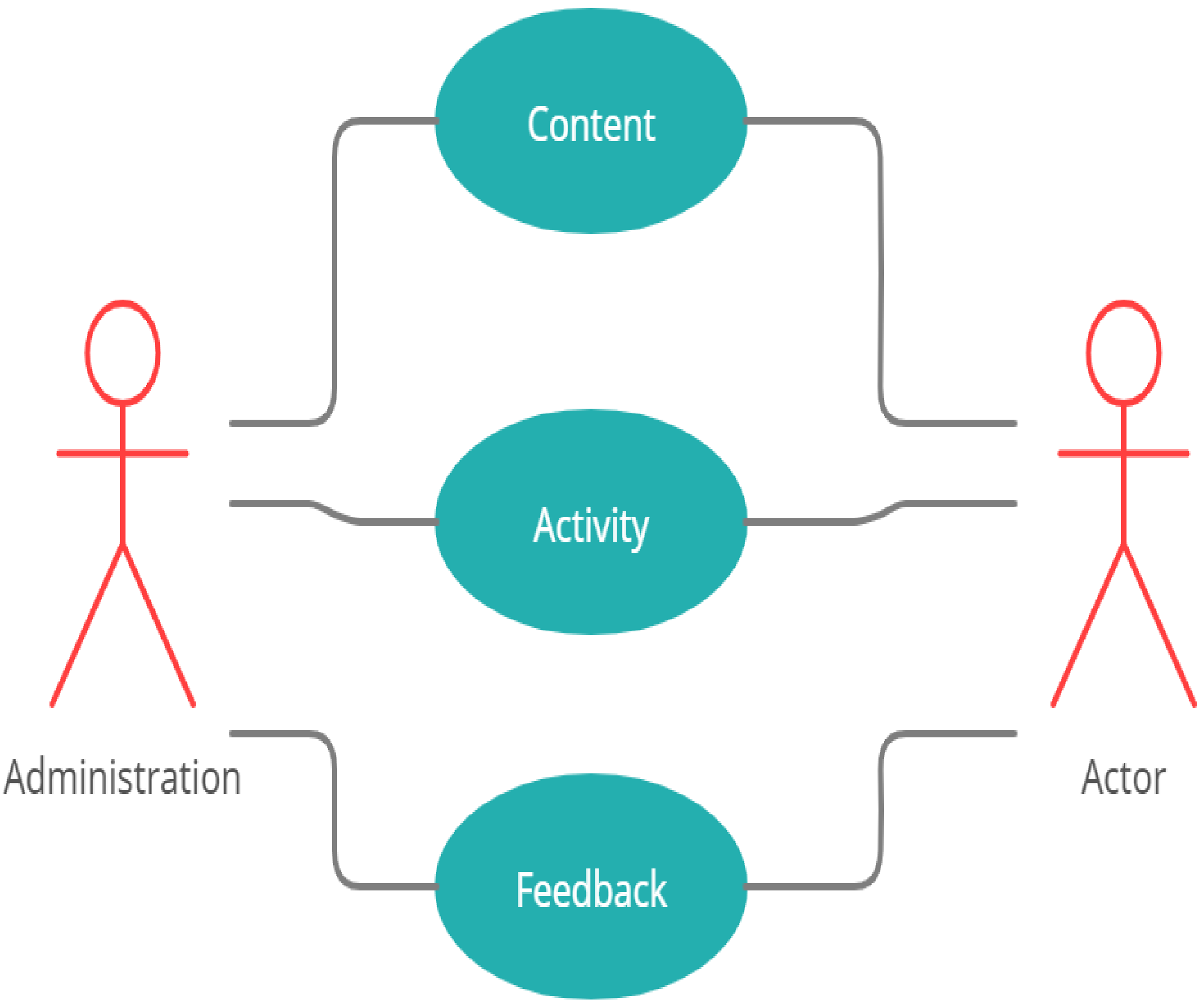


Fig 2. UML Diagram

This website will have only one actor i.e., user and it will require no authentication at this stage. User: can see the dashboard in which they can easily select the lesson and activity they can also give feedback and report any bug to developer. And Administrator is basically a developer who can add the content for now but in future enhancements we will make it open source so that developer and learner from everywhere can include their work and we will also include authentication for the same.

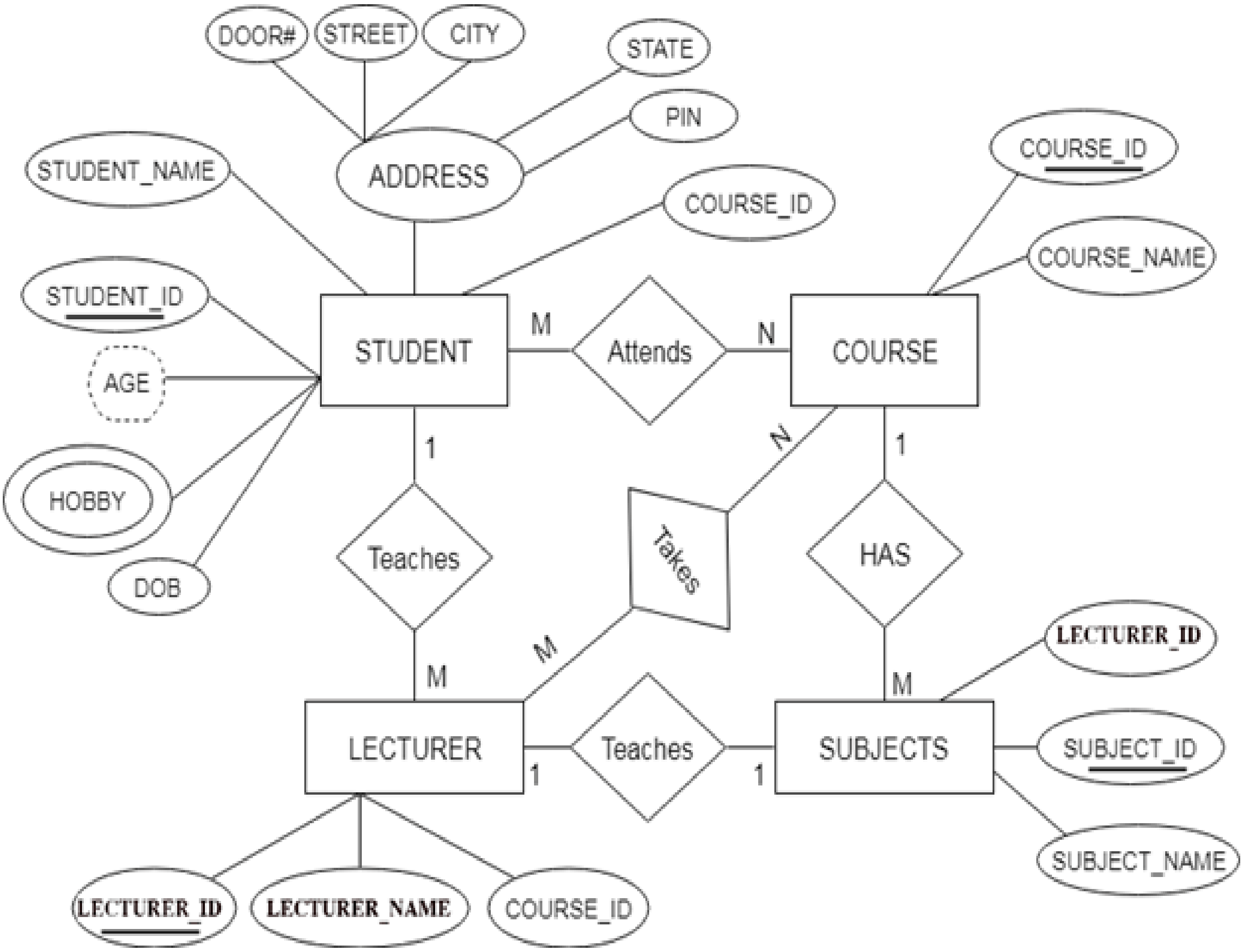


Fig 3. ER Diagram

This is the ER Diagram for the proposed system in which we have planed to implement a proper teaching learning concept and will get these data in our own database.

3.1 WORKING OF THE SYSTEM

Step 1: Start: With the start of the website show the Landing Page to the User where he can choose the lesson, activity or can give feedback or bug report.

Step 2: Making Choice of Lesson: After choosing the lesson the user will be redirected to the lesson content page of that corresponding lesson which can be visualize by and in the bottom of the page, he can go to the Major Activity of that corresponding lesson.

Step 3: Choosing Activity: User can also choose the activity of his choice from landing page as demonstrated and then user will be redirected to the activity page their choice.

Step 4: Interaction with Activity: When user will be redirected to the Activity page there they can interact with the activity and get to know the and understand the concept associated with that. And interactive activities will make their learning enthusiasm alive.

3.2 SAMPLE CODE SNIPPET

```
import React, { Component } from "react";
import PropTypes from "prop-types";
import Square from "./Square.js";

export default class Board extends Component {
  static propTypes = {
    cSquares: PropTypes.array.isRequired,
    qSquares: PropTypes.array,
    cycleSquares: PropTypes.array,
    cycleMarks: PropTypes.array,
    onSquareClick: PropTypes.func.isRequired,
    /**
      Passes index of square that was clicked up to Game.handleSquareClick.
    */
  };

  renderSquare(i) {
    return (
      <Square
        cMark={this.props.cSquares[i]}
        qMarks={this.props.qSquares[i]}
        onClick={() => this.props.onSquareClick(i)}
        isHighlighted={Boolean(
          this.props.cycleSquares && this.props.cycleSquares.includes(i)
        )}
        isBeingCollapsed={this.props.collapseSquare === i}
        cycleMarks={this.props.cycleMarks}
      />
    );
  }

  render() {
    return (
      <div>
        <div className='board-row'>
          {this.renderSquare(0)}
          {this.renderSquare(1)}
          {this.renderSquare(2)}
        </div>
        <div className='board-row'>
          {this.renderSquare(3)}
          {this.renderSquare(4)}
          {this.renderSquare(5)}
        </div>
        <div className='board-row'>
          {this.renderSquare(6)}
          {this.renderSquare(7)}
          {this.renderSquare(8)}
        </div>
      </div>
    );
  }
}
```

CHAPTER- 4
MODULE DESCRIPTION

A module description provides detailed information about the module and its supported components, which is accessible in different manners. The included description is available by reading directly, by generating a short html-description, or by making an environment check for supported components to check if all needed types and services are available in the environment where they will be used. This environment check could take place during registration/installation or during a separate consistency check for a component.

4.1 MODULE DESCRIPTION TABLE

Author	Siddharth Raj, Prerna Sharma
name	Quantum Learn
description	We will make all things available on a single website having information, resources, and interactive activities related to Quantum Computing which will help in learning and understanding quantum computing concepts in easy way. This will eventually make life easier and enhance learning enthusiasm of learners
loader-name	Webpack 5
supported-service	Web Application on browser >0.2% market share and last 2 version of any browser and not dead state.
service-dependency	Hosting it on Cloud Services like AWS, GCP
project-build-dependency	React Scripts: react scripts build; npm run build
runtime-module-dependency	“@testing-library/jest-dom": "^5.15.1", "@testing-library/react": "^11.2.7", "@testing-library/user-event": "^12.8.3", "assert": "^1.4.1", "bootstrap": "^5.1.3", "classnames": "^2.2.5", "prop-types": "^15.5.10", "react": "^17.0.2",

	"react-bootstrap": "^2.0.3", "react-dom": "^17.0.2", "react-router-dom": "^6.0.2", "react-scripts": "4.0.3", "web-vitals": "^1.1.2"
language	HTML, CSS, JavaScript
Status	Alpha, Beta Under Construction
Type	Web Application ReactJS

Table 1. Table of Module

4.2 Development Stage Module Descriptions

For every developmental stage, there is an expected developmental task. What happens when the expected developmental tasks are not achieved at the corresponding developmental stage? How can you help children achieve this developmental task?

How the proposed system looks like and will be defined and prepared from the requirement specifications that were analyzed and constructed.

Here Development means Web development technologies which refer to the multitude of programming languages and tools that are used to produce dynamic and fully-featured websites and applications.

It can be further divided into two parts such as Frontend Development and Back-End Development in this report we will further discuss these in brief:

Front-end (client-side) technologies.

Front-end technologies are for the “client side” of your website or application. They’re used to develop the interactive components of your website, and produce the elements that users see and interact with. This includes text colors and styles, images, buttons, and navigation menus.

Back-end (server-side) technologies.

Back-end technologies are for the “server side” of your website or application. They’re for developing the technical foundation. They store and arrange data and make sure everything on the front-end works. For example, when a user provides login credentials to a social media application, back-end technologies are used to check if those credentials are accurate. Once the credentials are verified, the server will send back the profile name, picture, and other associated information.

Back-end technologies are also used to streamline core business processes. In cases where you have lots of data that needs to be processed, you could run a script in the back-end to generate a meaningful report on the front-end. You can also send automatic emails to groups of users. Emails can be triggered by certain dates, such as the expiration of a user's free website trial.

We will further describe the use of these technologies in this report.

Frontend Development Technologies

1. **Dashboard:**

We design our Website Dashboards to be as easy to use as possible. In fact, if you are able to use email and any popular desktop publishing software, we are confident you will have no problem whatsoever using it. There is absolutely no need to know any code to be able to use your Website Dashboard and edit website content. And, if you do ever experience any difficulties and the built-in documentation does not help you can simply get in touch with us for free support.

Custom-Made & Easily Extended Because we build every website from scratch, your Website Dashboard will be tailor-made with only the features you need. And if you want us to add more features to your website in the future, your Dashboard will simply be upgraded to reflect this.

Like in the Fig. 4 below this is the landing page of the website i.e., dashboard which appears when any user opens the website. From here they can visit and redirected to different links and can do many things such as feedbacks.

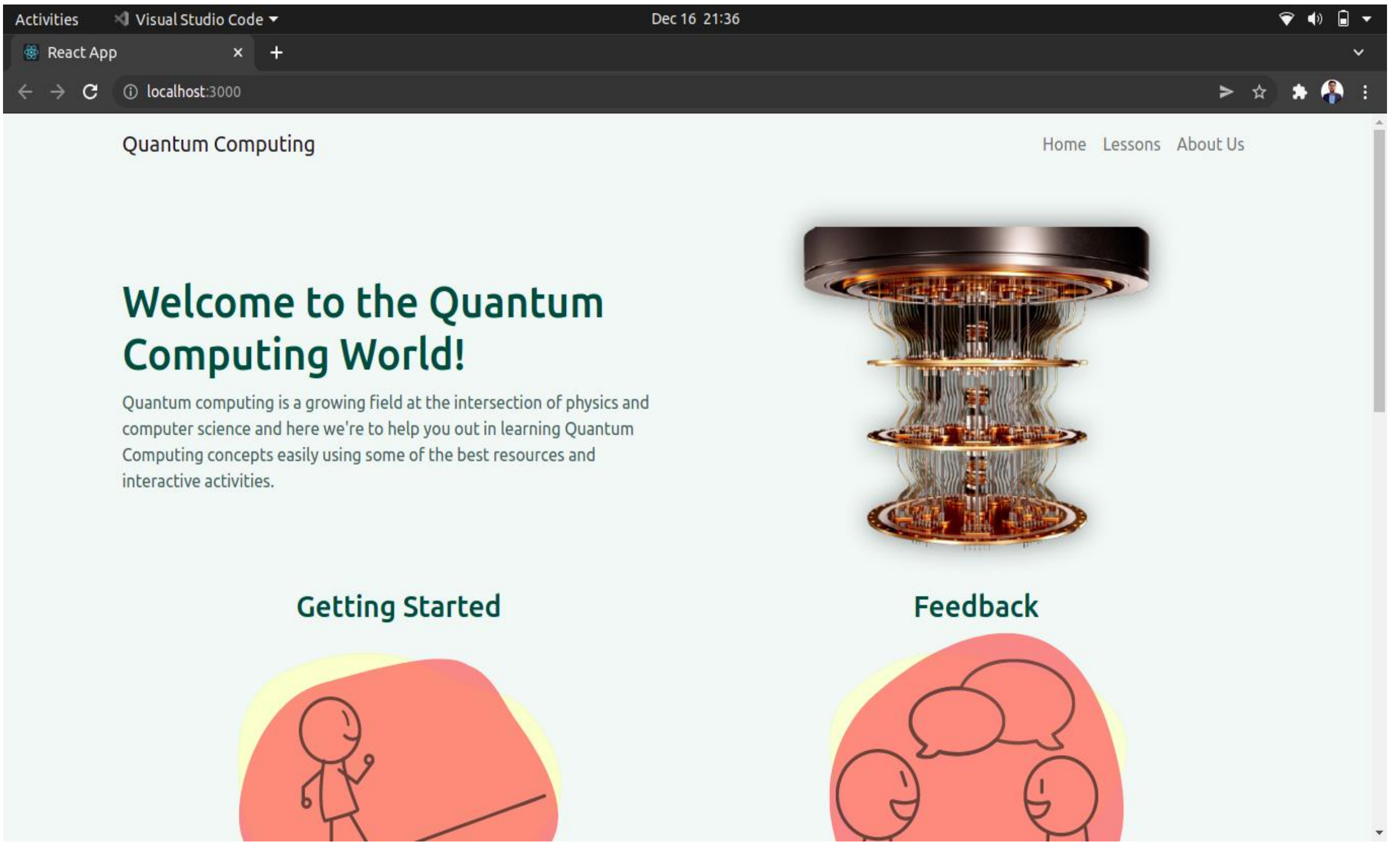


Fig 4. Dashboard

2. Navbar

On a website, a navigation menu is an organized list of links to other web pages, usually internal pages. Navigation menus appear most commonly in page headers or sidebars across a website, allowing visitors to quickly access the most useful pages.

Horizontal Navigation Bar

The horizontal navigation bar is the most common type of navigation menu. It lists the major pages side-by-side and is placed in the website header. Many websites feature the same sections, like “About,” “Products,” “Pricing,” and “Contact,” because visitors expect to see them. But these sections won’t necessarily be the most helpful to visitors on all sites.

The sections featured include three content categories — “News,” “Op-Eds,” and “Lifestyle” — as well as links to a submission page and a sign-up page. These are more likely to provide visitors with easy access to the pages they’re looking for rather than the standard About, Pricing, and Contact pages.

Dropdown Navigation Menu

Dropdown navigation menus are ideal for content-rich sites with a complex IA. If you’d like to include a lot of links to pages in your navigation bar, you can’t list them all side-by-side — it would either look cluttered or be impossible to fit them all horizontally. Instead, list the most important or general items in the top-level navigation bar and include the rest in a dropdown menu.

Hamburger Navigation Menu

The hamburger menu is most often seen in mobile web design. With this approach, the navigation items are often listed horizontally on larger screen sizes and collapse behind a hamburger button on smaller screen sizes. When visitors click on this three-line icon, a vertical drop-down or horizontal pop-out appears with the navigation links. This type of design is ideal for mobile apps or sites where real estate is limited

Vertical Sidebar Navigation Menu

With this menu type, the items are stacked on top of each other and positioned in the sidebar. While less popular than horizontal navigation, vertical navigation does offer several benefits. Since real estate isn't as limited, you can write longer navigation links and include more top-level options. It's also more eye-catching, which makes this style work well for agencies and other creative businesses.

Footer Navigation Menu

A footer menu is typically paired with — and expands upon — a horizontal navigation bar. If a visitor doesn't find the link they're looking for in the header, they can scroll down to the bottom of the page for more options.

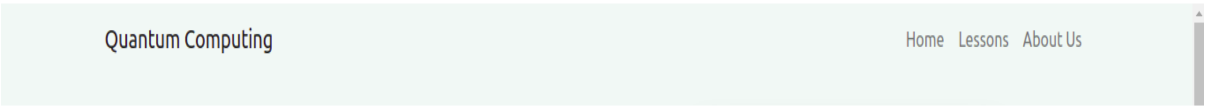


Fig 5. Navigation Menu

In Fig. 5 it is the Navbar of the website which will appear on the top of the website and user can navigate for different links from here.

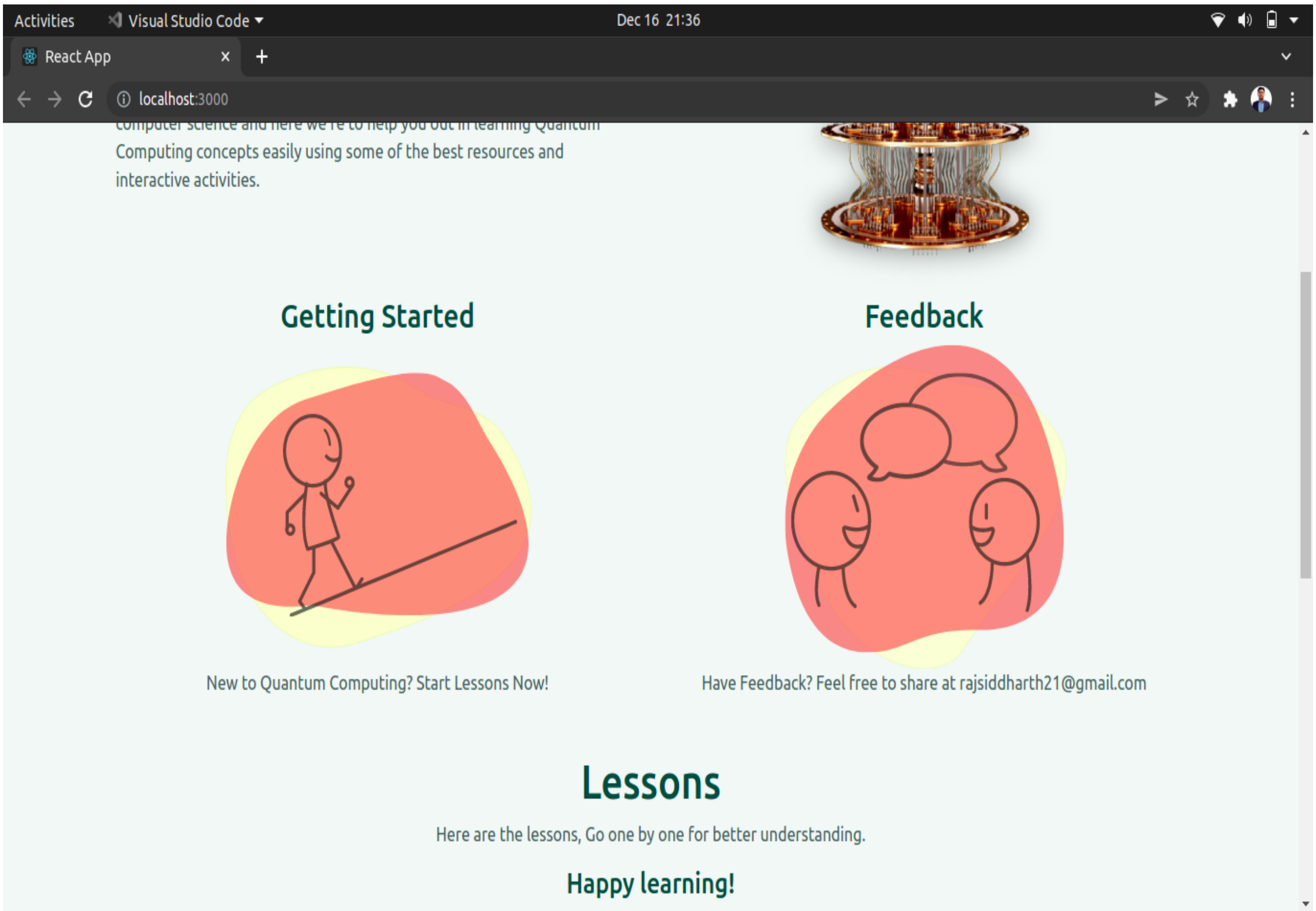

Activities
Visual Studio Code
Dec 16 21:36
React App
localhost:3000
Computing concepts easily using some of the best resources and
interactive activities.
Getting Started
Feedback
New to Quantum Computing? Start Lessons Now!
Have Feedback? Feel free to share at rajsiddharth21@gmail.com
Lessons
Here are the lessons, Go one by one for better understanding.
Happy learning!

Fig 6. Links

Link Page in Fig 6. It is the link page of the website which contains some useful links like lessons link and feedback link.

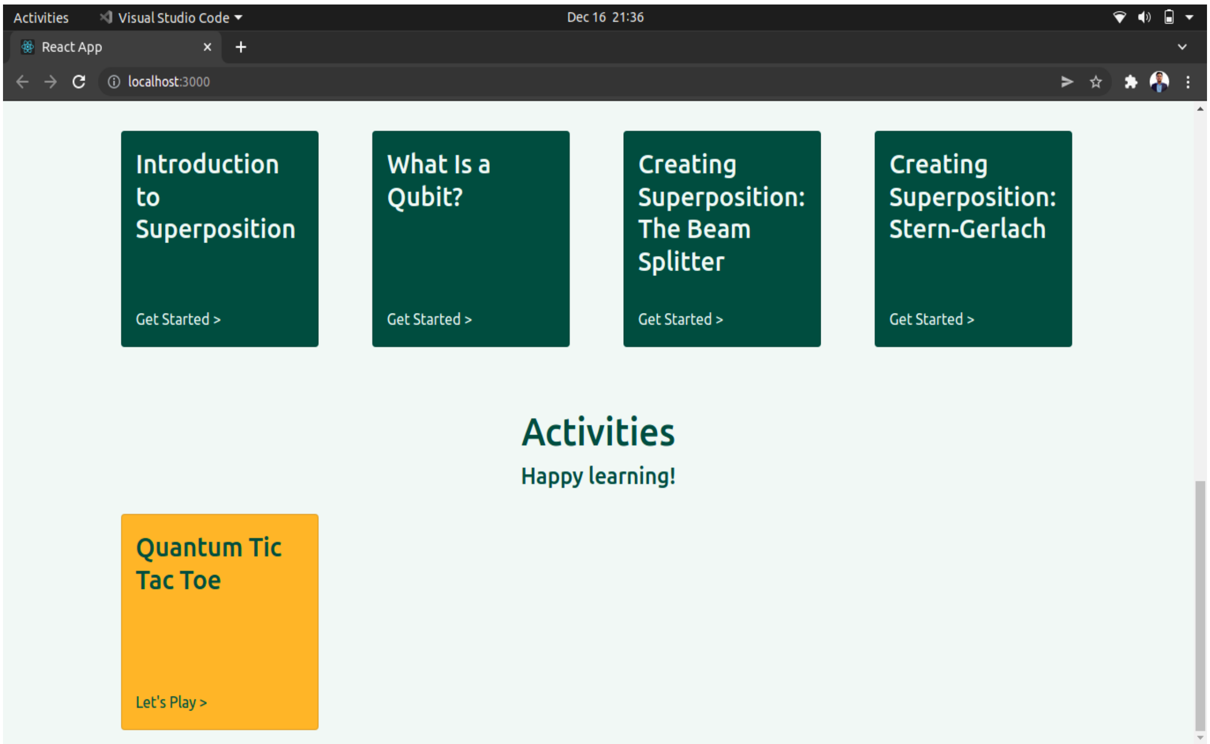


Fig 7. **Link to lessons and activities.**

In Fig. 7 these are links to the lessons which have content related to the Topic that is mentioned there and it also contains some interactive activities and the other card has link to the activities directly for better convenience.

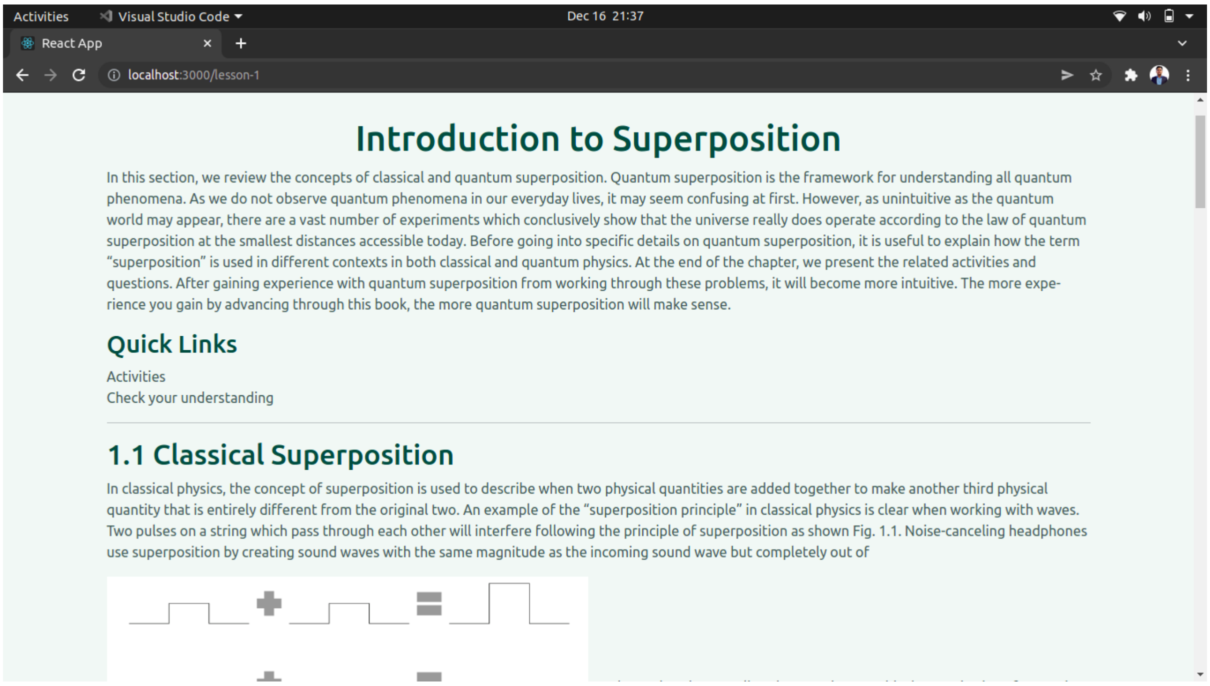


Fig 8. Content of the lesson

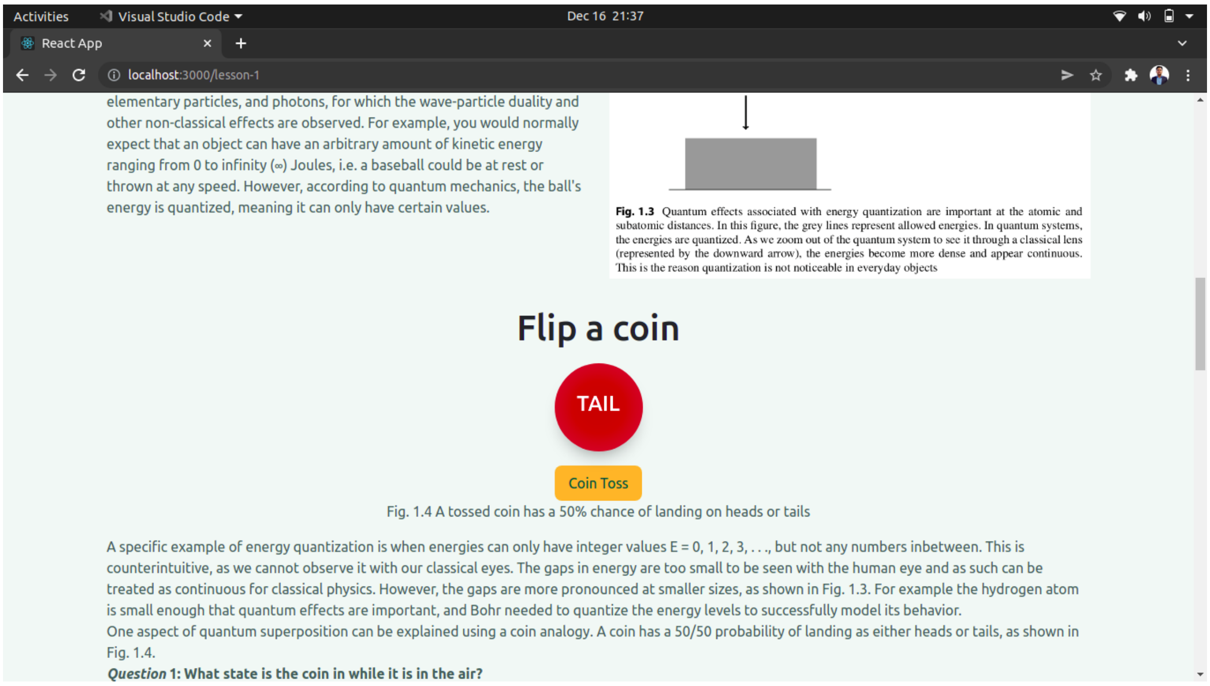
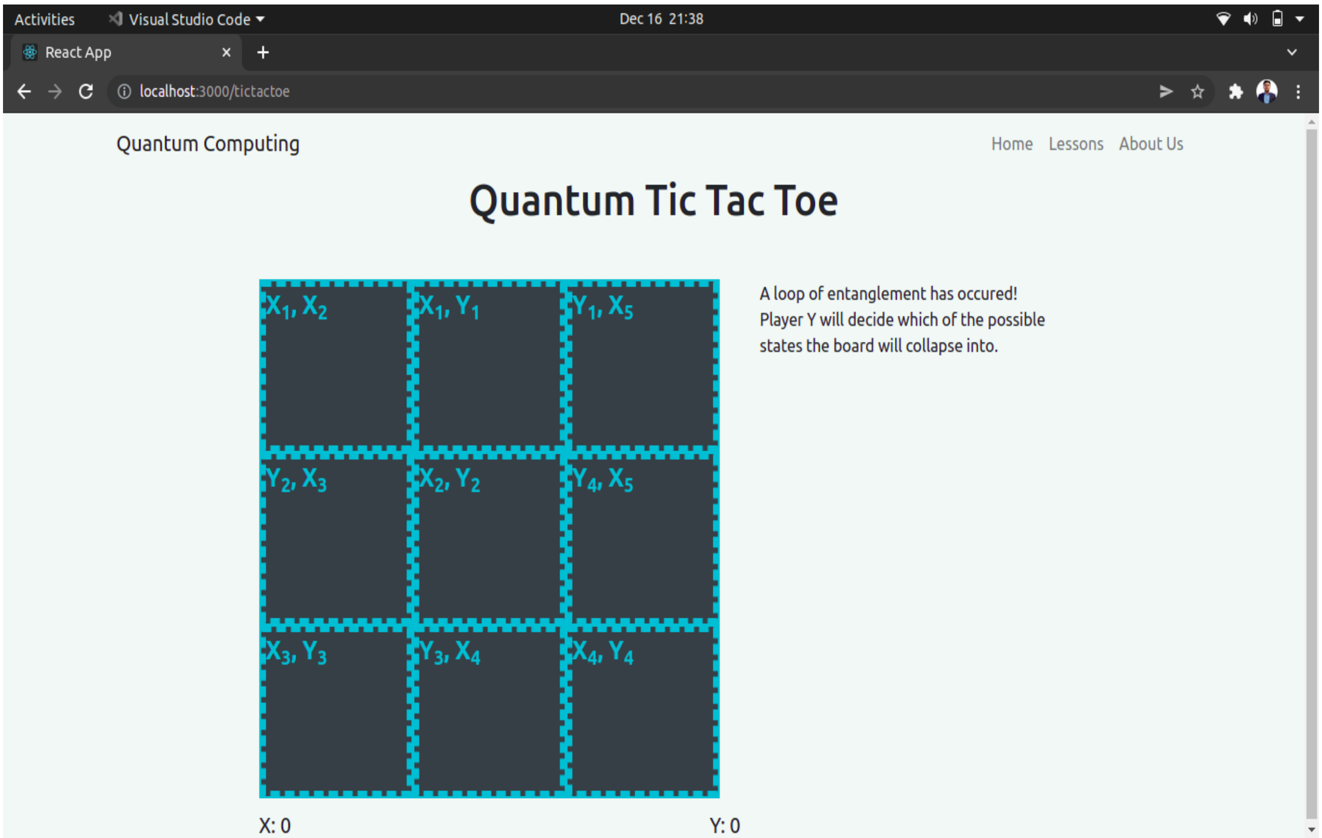


Fig 9. Minor Activity

In Fig 8. It represents the demo content of the lesson likewise all lessons contain their unique content relevant and to them and ensures that the content is appropriate and sufficient to understand the lesson and each lesson have their Major and some Minor Activity like in Fig. 9 It is the visual representation of one of the minor activities that are present in the website for lessons in this minor activity we can flip a coin which elaborates the concept of 50% of chance of getting heads and 50% chance of getting tails when you flip a coin. There are some of the really interesting major activities in lessons we will discuss them and refer to one of them in next figure.

Fig 10. Major Activity



In Fig 10. It is the visual representation of one of Major activities that are present in each lesson. This activity is called Quantum tic-tac-toe is a "quantum generalization" of tic-tac-toe in which the players' moves are "superpositions" of plays in the classical game. The game was invented by Allan Goff of Novatia Labs, who describes it as "a way of introducing quantum physics without mathematics", and offering "a conceptual foundation for understanding the meaning of quantum mechanics in this game instead of having single move at a time for each player like traditional tic-tac-toe game, in quantum tic-tac-toe each player has two moves which describes the concept superposition. We will discuss the algorithm of this Quantum Tic Tac Toe further in this report.

B. Algorithm of the System:

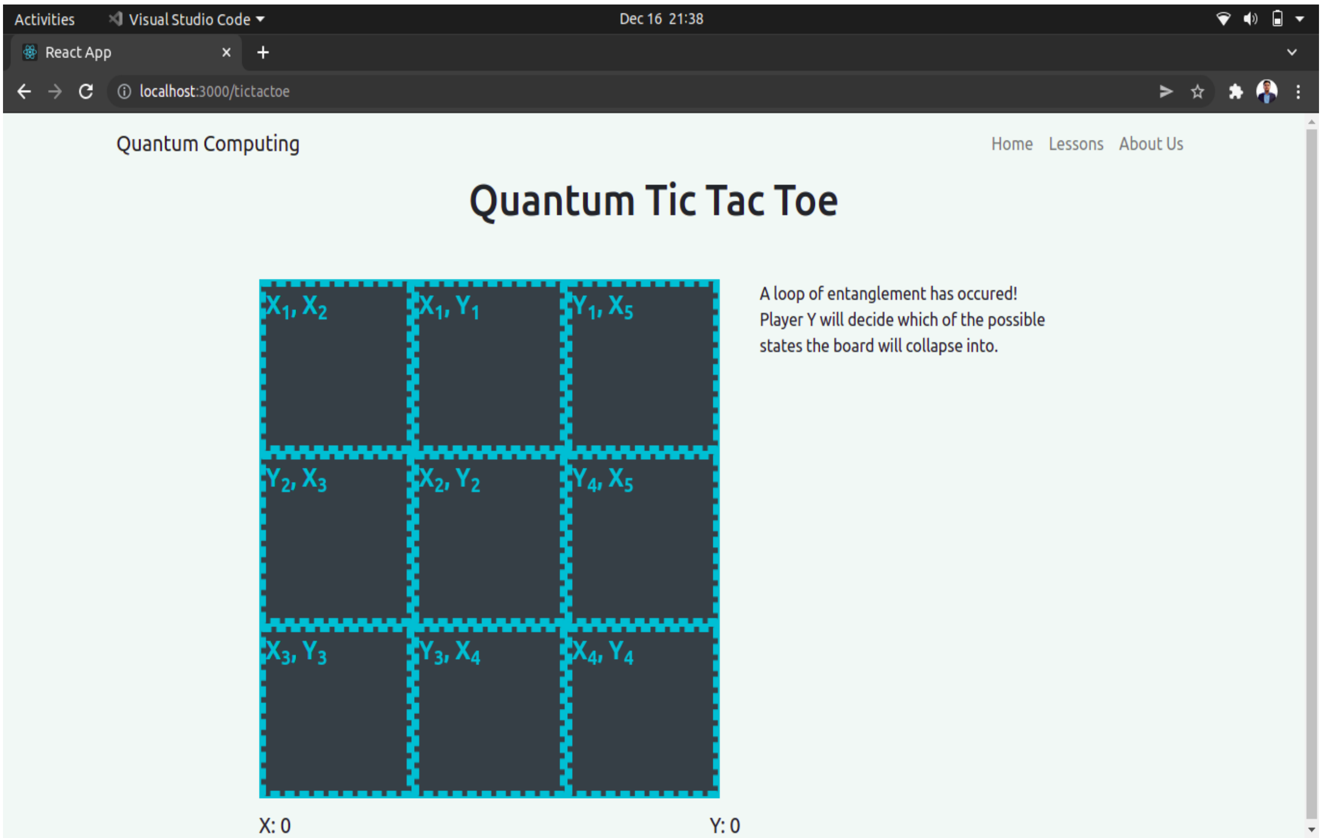
Step 1: Start: With the start of the website show the Landing Page to the User Fig 4. where he can choose the lesson, activity or can give feedback or bug report.

Step 2: Making Choice of Lesson: After choosing the lesson the user will be redirected to the lesson content page of that corresponding lesson which can be visualize by Fig 8. And in the bottom of the page, he can go to the Major Activity of that corresponding lesson.

Step 3: Choosing Activity: User can also choose the activity of his choice from landing page as demonstrated in Fig 7. And then user will be redirected to the activity page their choice.

Step 4: Interaction with Activity: When user will be redirected to the Activity page there they can interact with the activity and get to know the and understand the concept associated with that. And interactive activities will make their learning enthusiasm alive.

4.3 Quantum Tic Tac Toe



Alan Goff extended the popular game of tic-tac-toe based on the quantum physics principle of superposition, resulting in a game that is substantially more interesting and challenging: Quantum Tic-Tac-Toe. The rules are explained well, and are summarized in Understanding the game rules is essential for understanding our research.

Theorem for Generalization

[10]The rules for QT3 define entanglement using the notion of a cycle, which is difficult to generalize. Instead, we define entanglement to occur when there are some *x* pairs of pieces which are entirely contained within *x* squares. In this section, we prove the equivalence of these two definitions.

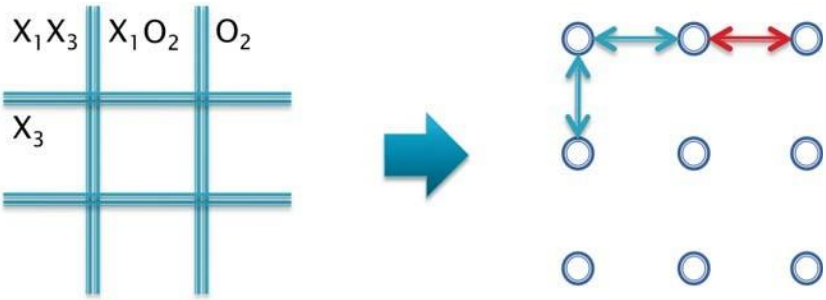


Figure 11: Example transformation of a QT3 game into a graph.

Consider a multigraph *G* with vertices corresponding to squares of the board. Draw an edge between 2 vertices for each pair of quantum moves placed on the 2 corresponding squares (Figure 1). By the game rules, the two endpoints of any edge in G are distinct, but a pair of vertices may be connected by more than one edge. By the rules of QT3 whenever a cycle appears on the board collapse will take place to remove it, thus without loss of generality we consider a graph that does not initially contain a cycle. With that, the equivalence of the two definitions can be stated as follows, with Goff's original definition as condition 1 and our definition as condition 3.

Theorem. *For any graph G which does not initially contain a cycle and in which edges are added one by one, the following conditions hold* for the first time *simultaneously:*

1. *G contains a cycle;*

2. *G contains exactly one cycle;*

3. *G contains a connected component with x edges and x vertices.*

Proof. We will use the fact that all trees (connected graphs with no cycles) have exactly one less edge than vertices [3].
1 $\Longrightarrow$ 2: If the last added edge AB produces two cycles $AC_1C_2 \ldots C_iBA$, $AD_1D_2 \ldots D_jBA$ *then* there must have been a cycle $AC_1C_2 \ldots C_iBD_j \ldots D_2D_1A$ before this edge was added, which is a contradiction. Hence G contains exactly one cycle.
2 $\Longrightarrow$ 3: Consider the maximal connected component $H \subseteq G$ that the cycle is in, which has v vertices and e edges. Then removing any edge of the cycle turns H *into* H', which has no cycles yet is still connected. Hence H' is a tree with v *vertices* and $e - 1$ edges, so $e = v$.
3 $\Longrightarrow$ 1: Let H be the stated connected component. If H contains no cycles, then H is a tree, contradiction. Hence H (and therefore G) contains a cycle.

This theorem is important and allows us to see that entanglement and collapse occur at the last possible moment, in the sense that a board with $x + 1$ pairs of pieces in x squares does not have a valid collapse by the Pigeonhole Principle. Furthermore, this definition of entanglement can be very easily generalized, as presented in the next section.

Generalized Rules of the Game

After formulating and proving the important theorem in the previous section, we now present the rules of our *Generalized QT3* (GQT3).

Two players first agree on 2 integers $m, n > 0$. The game is played on an $m \times m$ squareboard, where each square is either quantum or classical, with all squares initially quantum. Each turn, a player places n copies of the turn number (collectively called a *series*) on n quantum squares of his choice. (If there are less than n quantum squares, classical tic-tac-toe is played.) The maximal group of x squares that completely contains x series is considered entangled.

When a player causes entanglement, the other player collapses the board by fixing each of the x series part of the entanglement to one of the n squares that it occupies, such that each of the x squares is assigned to exactly one series. Then the series become classical and replace the quantum pieces in their respective squares, and all of the x squares become classical.

The game ends when all squares are classical and hence occupied by exactly one number. Points are then awarded to players with lines (rows, columns or diagonals) filled only with their numbers. The i^{th} line obtained (ordered by the maximal number in the line) is awarded point. Note that GQT3 with $m = 3, n = 1$ is normal Tic-Tac-Toe, and $m = 3, n = 2$ is equivalent to QT3, as shown in the previous section.

Proof of Consistency

When entanglement occurs, it is not obvious whether a collapse is always possible. In thissection, we prove the consistency of the game by showing that there must always exist *many* collapses.

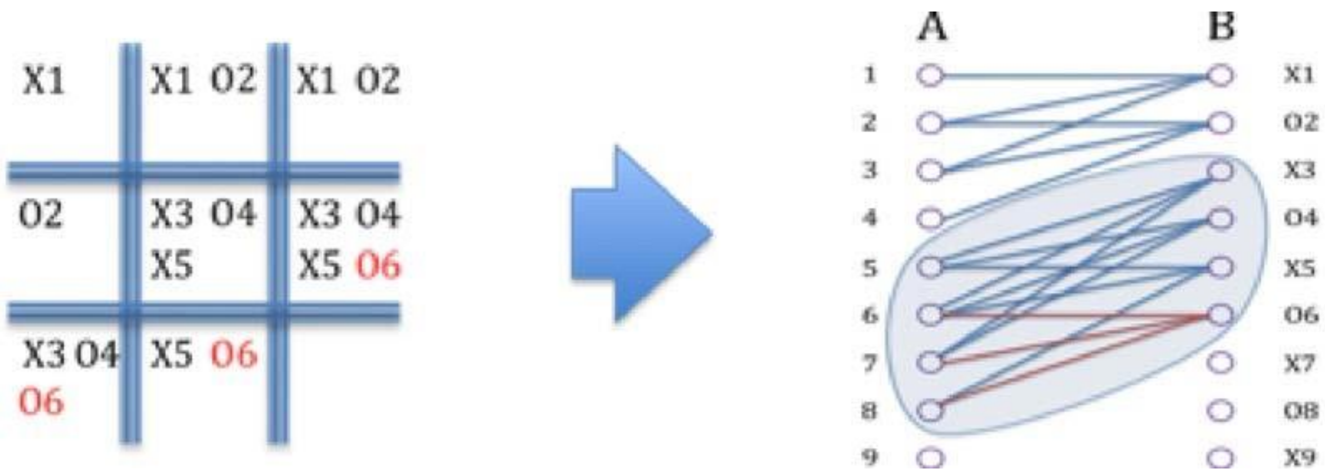


Figure 12: An example bipartite graph G, for $m = 3$, $n = 3$. Note that collapse corresponds to choosing a perfect matching on G.

Represent the GQT3 board as a bipartite graph G with bipartition A and B, with A represent- ing the squares of the board and B representing the series of pieces (Figure 2). When entanglement occurs, consider the maximal $E \in B$ *such* that $|n(E)| \leq |E|$. Then the graph G' induced on the vertices in $E \cup n(E)$ represents the entangled portion of the game board, and satisfies the following:

1. If for any nonempty $S \subseteq E$, we have $|n(S)| < |S|$, then entanglement would have occurred earlier in $S \cup n(S)$, contradiction. Thus $|n(S)| \geq |S|$. In particular, $|n(E)| = |E|$.
2. If this is the first entanglement of the game then $\deg(v) = n$ for all vertices $v \in E$, since each quantum series contains exactly n pieces.

   Else, if there exists $v \in E$ such that $\deg(v) = 1$ then consider the board at the previous entanglement, say on $E_0 \cup n(E_0)$. Note that $|n(E_0 + v)| = |n(E_0) \cup n(v)| = |E_0 + v|$, so the previous entanglement should have been on $(E_0 + v) \cup n(E_0 + v)$ instead of $E_0 \cup n(E_0)$ by maximality, contradiction. Hence $\deg(v) \geq 2$ for all $v \in E$.

Theorem. *Let* $k_0 \in \mathbb{N}$. *A bipartite graph* G' *with bipartition* A, B *is given such that:*

1. $|A| = |B|$;
2. *For all* $S \subseteq B$, *we have* $|n(S)| \geq |S|$; *and*
3. *For all vertices* $v \in B$, $\deg(v) \geq k_0$. *Then*

   G' *has at least* $k_0!$ *perfect matchings.*

Proof. We proceed by induction on k_0. For $k_0 = 1$ the theorem holds by Hall's Marriage Lemma. Assume induction hypothesis for some $k_0 = k \geq 1$, and let graph G satisfy the premises of the induction hypothesis for $k_0 = k + 1$. We then choose the smallest $S_0 \subseteq B$ such that $|n(S_0)| = |S_0|$.

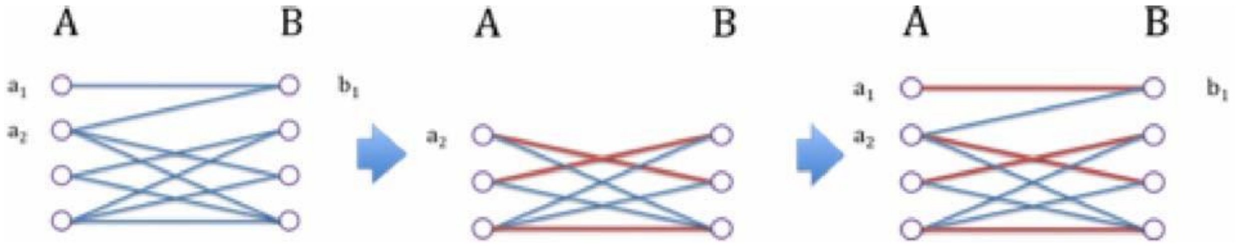


Figure 13: Left: Original graph. Center: After edge removal. Right: After adding back the edges.

This subset exists because $|n(B)| = |A| = |B|$. For any nonempty $S \subset S_0$, $|n(S)| > |S|$ by the minimality condition on S_0.

Select an arbitrary vertex b_1 from S_0 and $a_1, a_2, \ldots, a_{k+1} \in n(b_1) \subseteq n(S_0)$.

Now remove b_1 and a_1, along with all edges incident to them. Now for all $S \subset S_0$, $|n(S)| \geq |S|$ since only one vertex was removed from $n(S_0)$, and hence at most one vertex was removed from $n(S)$. Similarly, for all $b \in S_0$, $\deg(b) \geq k$. Hence the inductive hypothesis states that there are at least $k!$ perfect matchings on $S_0 \cup n(S_0)$.

Combined with edge (a_1, b_1), this means that there are at least $k!$ perfect matchings on $S \cup S_0$ with b_1 matched to a_1. Since we can repeat the argument for any edge from (a_1, b_1) to (a_{k+1}, b_1), there are at least $(k+1) \cdot k! = (k+1)!$ perfect matchings on $S \cup S_0$.

Let $X = B \setminus S_0$, and $T = A \setminus n(S_0)$. For any $S \subseteq X$, $n(S \cup S_0) \subseteq (n(S) \cap T) \cup n(S_0)$. Hence if $|n(S) \cap T| < |S|$ then

$$|n(S \cup S_0)| \leq |n(S) \cap T| + |n(S_0)| < |S| + |S_0| = |S \cup S_0|,$$

where the last equality holds because $S \subseteq X$ and S_0 are disjoint. This contradicts condition 2 of the induction hypothesis.

Therefore $|n(S) \cup T| \geq |S|$ for all $S \subseteq X$, so there is a perfect matching on $X \cup T$ by Hall's Marriage Lemma. Hence there are at least $(k+1)!$ perfect matchings on the original graph G, and induction is complete. $\square$

Hence by applying the theorem on G', there are always ≥ 2 collapses possible. Further, at the first entanglement there are always $\geq n!$ collapses.

Entanglement Detection

The discussion in the previous section shows that a large number of collapses are possible when entanglement occurs on a GQT3 board. However, nothing has been mentioned about *how* to determine when an entanglement has occurred! In this section, we describe an efficient polynomial- time algorithm to determine for any GQT3 game whether there is an entanglement on the board.$n(S)$ denotes the set of vertices adjacent to at least one vertex of S.

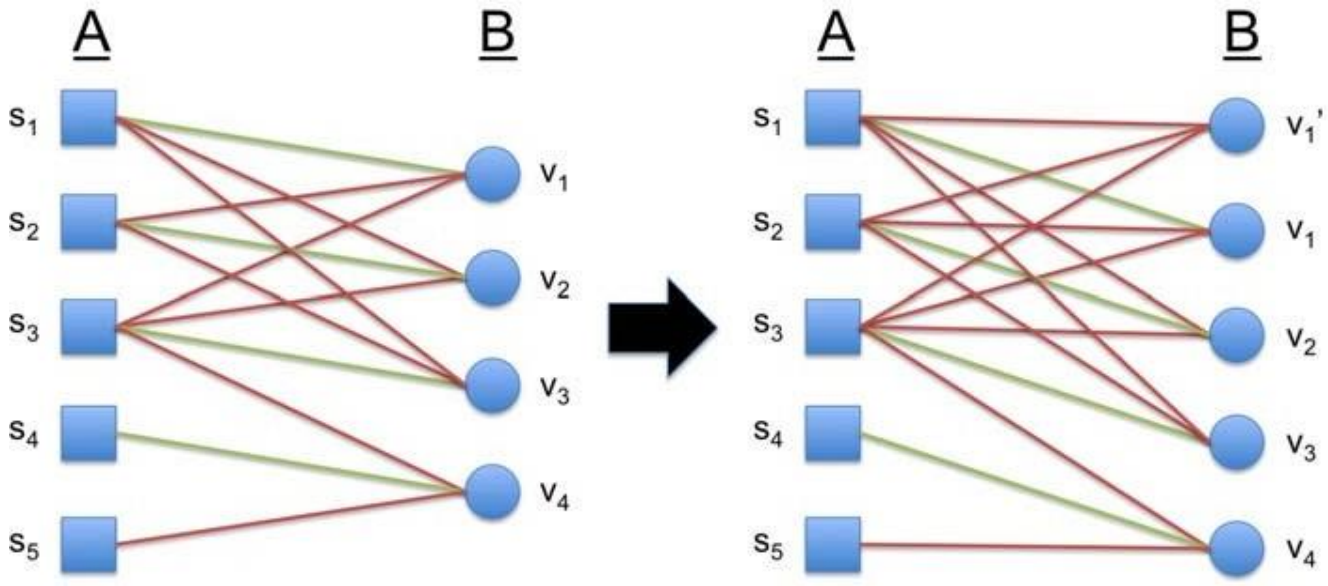


Figure 14: Example G (left) and G_v (right), with $v = v_1$. Maximal matchings are highlighted in green.

Then

For any GQT3 game board, construct the bipartite graph G as in the previous section, with bipartition A (representing squares) and B (representing series). By a similar argument in the previous section, we deduce that $|n(S)| \geq |S|$ for all subsets $S \subseteq B$. This implies by Hall's Marriage Lemma that there exists a matching in G with size $|B|$, which is clearly maximal.

Choose any vertex $v \in B$, and suppose v is adjacent to vertices $a_1,..., a_k$ in A. We add a new vertex v' to the graph, and add new edges $v'a_1,..., v'a_k$, calling the resulting graph G_v. In effect, v' is a *duplicate* of v. Also, define $S_v = S + v'$ for any $S \subseteq B$.

We find the size of the maximal matching of G_v, by considering the following two cases.

$E = \{v_1, v_2, v_3\}$ satisfies the conditions in Case 1.

Case 1: *There exists a subset $E \subseteq B$, with $v \in E$, such that* $|n(E)| = |E|$, i.e. there is an entanglement on the board containing v (Fig. 1). Then by the above condition there must exist a perfect matching on $E \cup n(E)$, by Hall's Marriage Lemma.

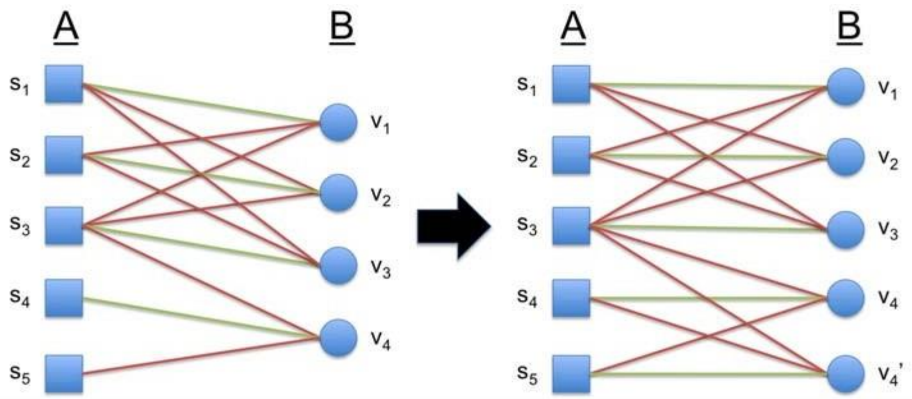


Figure 15: Example G (left) and G_v (right), with $v = v_4$. Maximal matchings are highlighted in green.

Consider the maximal matching on G_v. This matching has size at least $|B|$, since G is a subgraph of G_v. Moreover, note that $|n_{GV} (E_v)| = |n_G(E)| = |E| = |E_v| - 1$. Hence there does not 5 exist a matching in G_v of size $|B_v| = |B| + 1$. Thus, the maximal matching on G_v contains exactly $|B|$ edges.

Case 2: *For all subsets $E \subseteq B$, with $v \in E$, we have* $|n(E)| \geq |E| + 1$, i.e. there is no entanglement on the board containing v (Fig. 2). Consider some $S \subseteq B_v$.

1. If at most one of v, v' is in S *then* note that we may swap v *and* v'; hence we may assume without loss of generality that $v' \notin S$, so $S \subseteq B$. Then $|n_{G_V}(S)| = |n_G(S)| \geq |S| + 1$.
2. If both v, v' are in S *then* $|n_{G_V}(S)| = |n_G(S - v')| \geq |S - v'| + 1 = |S|$.

Thus, the maximal matching on G_v contains $|B_v| = |B| + 1$ edges by Hall's Marriage Lemma.

The above discussion implies that *the size of the maximal matching in G_v is*

$$\begin{cases} |B| & \text{if } v \text{ is contained in an entanglement;} \\ |B| + 1 & \text{otherwise.} \end{cases}$$

This suggests a quick algorithm to determine the existence of entanglement in G: for all $v \in B$, compute the size of the maximal matching for G_v using well-known polynomial-time algorithms (e.g., the Hopcroft-Karp algorithm). The vertices v for which this size is $|B|$ constitute exactly the set-in which entanglement occurs. In particular, if all the computed sizes are equal to $|B| + 1$ then entanglement has not yet occurred on the board.

Game Tree Size

The game tree size is the total number of possible games that can be played, or the number of leaves in the game tree. As we are not interested in determining a winner when counting the game tree size, the number of squares in the board does not need to be a perfect square and we define s to be the number of squares ($s \in \mathbb{N}$). It turns out that computing the game tree size through a brute force search in the game tree would take approximately a year and is hence not feasible.

The crux moves in solving this conundrum is realizing that the number of move combinations that result in any particular endgame is the same. This is because we can simply rearrange the squares to transform one endgame and its move combinations into another. Thus we only need to count the number of ways to achieve a certain specified endgame and then multiply that value by $s!$. We choose to compute the number of moves to achieve an endgame where the classical piece i is on the i^{th} square of the board (Figure 4). To do this, on the i^{th} turn, we affix one piece of the i^{th} series on the i^{th} square and perform an exhaustive search for all possibilities for the other $n-1$pieces. Whenever collapse occurs, we prune out possibilities that cannot end in the required end game state. For example, we ignore cases where the piece 3 (X_3) lands up on square 2, since we want it to land on square 3. Our Java implementation of this runs in about 15 seconds for $n = 2$,

$s = 9$ and yields an enormous value of about 18.5 trillion (18,539,269,580,160). As a comparison, this is roughly 73 million times the game tree size of classical tic-tac-toe, which is 255,168. This provides mathematical evidence for our intuition that GQT3 is a very challenging game that is difficult to play and analyze.

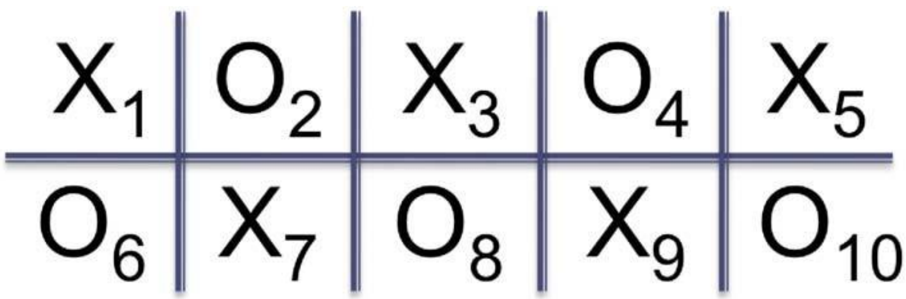


Figure 16: The end game state we use to compute game tree size

We can also estimate bounds for the game tree size. $\binom{s}{n}^{s}$ is an upper bound since at most s squares can be quantum at any turn. $\binom{s}{n}\binom{s-1}{n}\binom{s-2}{n}\cdots\binom{n}{n}$ is a lower bound since at least $s-t$ squares will be left quantum after t moves. For the case $n = 2$, we can do even better using our symmetry idea mentioned above. Using a similar method, we obtain a lower bound of $s! \times (s-1)!$ and upper bound of $s! \times (s-1)^s$. Table 1 shows the upper bound, correct value and lower bound of the game tree size for $n = 2$, $1 \le s \le 9$.

s	1	2	3	4	5	6	7	8	9
Upper	1	2	48	1940	1.23×10^5	1.13×10^7	1.41×10^9	2.32×10^{11}	4.87×10^{13}
Actual	1	2	42	1370	7.33×10^4	5.86×10^6	6.53×10^8	9.71×10^{10}	**1.85×10^{13}**
Lower	1	2	12	144	2.88×10^3	8.64×10^4	3.63×10^6	2.03×10^8	1.46×10^{10}

Table 2: Values of the Game Tree Size for *n* =2

1. **Random Play**

An interesting situation is a GQT3 game where both players play their moves randomly. In Section 5 we explained that there is a bijection in the move combinations resulting in any 2 distinct endgames. Thus, in a random game the probability of any endgame occurring is equal. Therefore, we can analyze this scenario by analyzing the distribution of the endgames. In a 2-by-2 board, player 1 will always win by 0.5 points since this is the result in every endgame. *m* = 3 (3-by-3 board) is more interesting. There are 362,880 endgames which can be distributed into the player 1-win margins as in table 2 below. These values were computed combinatorically and verified using a computer program. Using this we can calculate the probability of each win margin

state occurring. *

Type	Player 1 Win Margin						
	-1	-0.5	0	0.5	1	1.5	2
Number of Endgames	34,560	69,984	46,080	33,696	115,200	50,688	12,672
Probability	9.5%	19.3%	12.7%	9.3%	31.7%	14.0%	3.5%
Total	28.8%		12.7%	58.5%			

Table 3: Analysis of Random Games for the 3-by-3 board

From this we can see that the most likely result (mode) is player 1 wins by 1 point. However, player 1 is expected to win the game by 0.452 (3sf) points.

1. **Perfect Play**

Even more interesting than random play is perfect play, where the 2 players choose the best move possible with the assumption that the other player will also play the best move possible. Most interesting in this analysis is the end result, the Nash Equilibrium [4]. †

We computationally determine the Nash Equilibrium, using the minimax algorithm as a skeleton. Without optimizations it would take about a year to traverse the huge game tree of

Goff's QT3. Thus, we use alpha-beta pruning [5], memoization (using a hash table), and symmetry considerations (rotation and reflections) to bring down the run time to under an hour. Through this method, we find the Nash Equilibrium of Goff's QT3 is a Player 1 win by 0.5 points.

This is unfair, thus we attempted to tweak the game rules. If we subtract 1 from the subscripts of all player 2's pieces, then the Nash Equilibrium is a draw. Also, if player 2 chooses collapse then the game is a draw. On the other hand, if player 1 chooses collapse, then he wins by 2 points, the largest possible margin. If the person who causes collapse chooses it then the Nash Equilibrium is a Player 1 win as well.

Unfortunately, for generalized m and n, the game tree is too large for such searches, hence finding the Nash Equilibrium in these cases is room for further research.

CHAPTER – 5
SYSTEM REQUIREMENTS

5.1 VISUAL STUDIO CODE

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas.

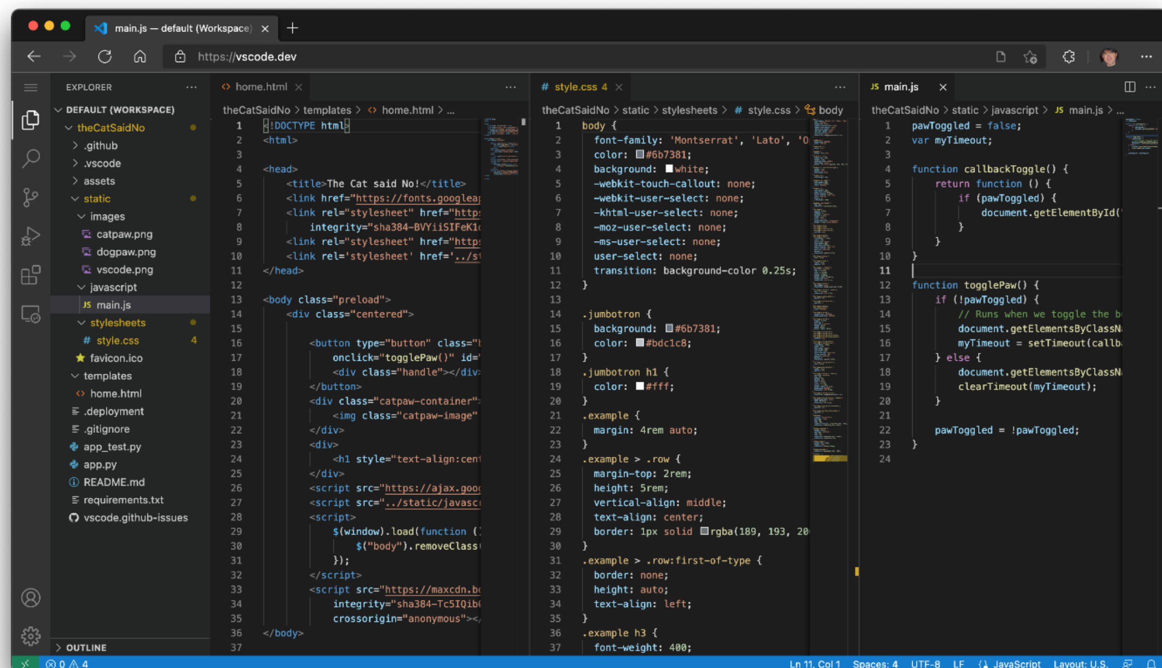


FIG 17: VSCODE IDE

5.2 HTML

The **HyperText Markup Language**, or **HTML** is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as <**img** /> and <**input** /> directly introduce content into the page. Other tags such as <**p**> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997. A form of HTML, known as HTML5, is used to display video and audio, primarily using the <**canvas**> element, in collaboration with javascript.[11]

5.3 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.[3] This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.[4]

The name *cascading* comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) `text/css` is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.[12]

5.4 Javascript

JavaScript,[13] often abbreviated **JS**, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. Over 97% of websites use JavaScript on the client side for web page behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on the user's device.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.

JavaScript engines were originally used only in web browsers, but are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js.

Although Java and JavaScript are similar in name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

5.5 ReactJS

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. However, react is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality[14]

Declarative

React makes it painless to create interactive UIs. Design simple views for each state in your application, and react will efficiently update and render just the right components when your data changes. Declarative views make your code more predictable and easier to debug.

Component-Based

Build encapsulated components that manage their own state, then compose them to make complex UIs. Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.

Learn Once, Write Anywhere

We don’t make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code.

React can also render on the server using Node and power mobile apps using React Native.

CHAPTER 6
RESEARCH PAPER

Here are some snapshots of research paper for your reference

UNDERSTANDING QUANTUM COMPUTING CONCEPTS BY ACTIVITIES USING WEB APPLICATION

SIDDHARTH RAJ
School of Computing and *Engineering*
(Undergraduate Student)
Galgotias University
(University)
Uttar pradesh,India

PRERNA SHARMA
School of Computing and *Engineering*
(Undergraduate Student)
Galgotias University
(University)
Uttar pradesh,India

DR. JAYAKUMAR V.
School of Computing and *Engineering*
(Assistant Professor)
Galgotias University
(University)
Uttar pradesh,India

Abstract— **The sole and main idea behind this project is that Quantum Computing is emerging field and people are still struggling to find proper resources to learn quantum computing concepts and the activities based learning is really very limited. Taking an example if someone wants to start learning or understanding quantum computing and how it works what are the concepts they have very limited options either they can go for books or they can refer to some online documentation activity based learning in this field is still missing or very minimal.There are some of resources that are available but all are scattered you can't find anything in just one place you have to go through many of the websites and books that are available to understand these concepts. We decided to implement this Idea in form of website so that any user can easily understand some working concepts of quantum computing all at one place with proper documentation and activities and sorted in a proper manner. For website we use HTML and, CSS for designing and ReactJS as framework, NodeJS for backend and MongoDB for database. This website will make their life easier who wants to learn concepts of quantum computing by providing all resources at one place and they don't need to spend time on searching resources.**

Keywords- quantum computing, HTML, CSS, ReactJS, NodeJs, MongoDB.

I. INTRODUCTION

Keeping in mind the emerging field of quantum computing and contribution and interest of people in that the demand of resources to learn quantum computing is increasing day by day.

Generally, People prefer learn by activities and in many of research and test it is found that learning by activities is very effective. But the resources to learn quantum computing concepts by activites is very limited and to fulfill the demand and keep the enthusiasm of learner alive we came with a solution to make a dynamic website using HTML, CSS,

Bootstrap and ReactJS which will help in understanding the Quantum Computing concepts with the help of some interactive activities.

There are some resources which are available on internet and also in books but they all are scattered anyone who want to learn something they have search things here and there and then they will get proper content with some activities and explanation and between all these there are a lot of waste of time.

We will make all things available on a single website having information, resources, and interactive activities which will help in learning and understanding quantum computing concepts in easy way.

This will eventually make life easier and enhance learning enthusiasm of learners.

II. IN TODAY'S SYSTEM

In today's system getting all resources for learning or understanding any concept through books or online resources is quite tough because you have to find the whole content and grab them from different resources and wrap them then only you will get proper understanding and understanding by doing activities is very rare because most of them are only way to interact with the activitiy only there is very bare minimum amount of content is available.

And when it comes to quantum computing, as it is an emerging field a lot of people shows interest in it but unfortunately the resources are limited and activities are also very limited most of them have only activites and missing the concepts behind that, one have to first go through the concepts from different resource and then do the activity on another resource or you may say another website and all of these are time taking tasks and will eventualy dim the enthusiasm of learning new concepts.

III. LITERATURE SURVEY

This section explores literature review of this system and discuss some of similar available systems.

There are some resources available on Internet and also in form of books some of them are :

1. Quantum Atlas [1] :
   It is a website for studying quantum technologies which also addresses some concepts of quantum computing, Qubits, Superposition the represtation and explanations available are very good they're explaining things using some beautiful illustrations you can visit the website on https://quantumatlas.umd.edu/ .
2. Books
   There are plenty of books available which ellaborate the concepts using images like "Quantum Computing for the Quantum Curious"[2] by Ciaran Hughes, Joshua Isaacson, Anastasia Perry, Ranbel F. Sun, Jessica Turner.

But the main issue that arises that these are not that interactive you'll get the theoratical knowledge but for actual interaction you have to go some other resources.

IV PROPOSED SYSTEM

This section explores the system we're proposing to make users life easy.

The most important steps that have been taking to build the we application for learning quantum computing with the help of activities are:

A. Initial Stage:
   Identifying the content and concepts that we have to include in our web application and markdown all activities that are necessary and we have to include them in our web application.

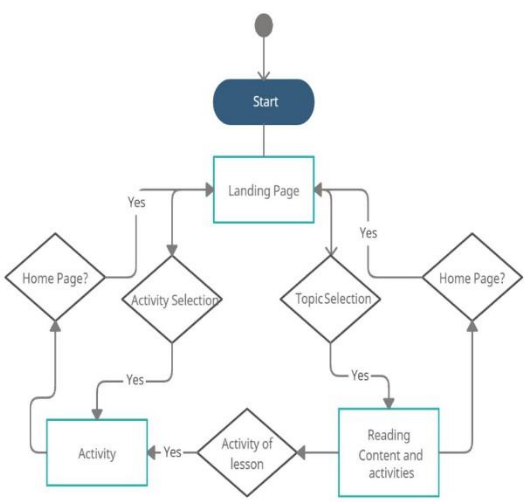


Fig 1. Architectural Diagram.

Actors and Modules of the system-

This website will have only one actor i.e. user and it will require no authentication at this stage.

User: can see the dashboard in which they can easily select the lesson and activity they can also give feedback and report any bug to developer.

B. Design Stage:
   How the proposed system looks like and will be defined and prepared from the requirement specifications that were analyzed and constructed.

Fig 2. Dashboard

Like in the Fig. 2 this is the landing page of the website i.e. dashboard which appears when any user opens the website.

Fig 3. Navbar

In Fig. 3 it is the Navbar of the website which will appear on the top of the website and user can navigate for different links from here.

Fig 4. Link Page

In Fig 4. It is the link page of the website which contains some useful links like lessons link and feedback link.

Fig 5. Link to lessons and activities.

In Fig. 5 these are links to the lessons which have content related to the Topic that is mentioned there and it also contains some interactive activities and the other card has link to the activities directly for better conveinience.

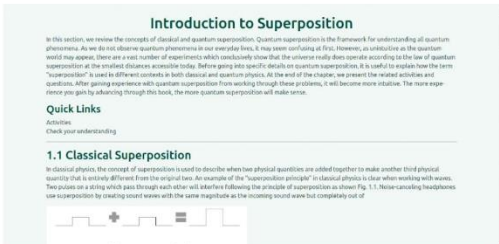


Fig 6. Content of the lesson

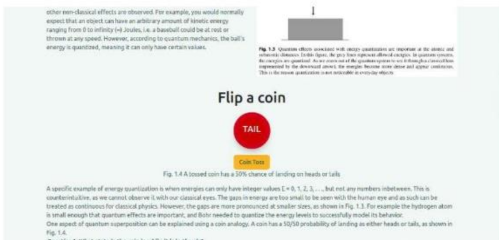


Fig 7. Minor Activity

In Fig 6. It represents the the demo content of the lesson likewise all lessons contain their unique content relevant and to them and ensures that the content is appropriate and sufficient to understand the lesson and each lesson have their Major and some Minor Activity like in Fig. 7 It is the visual representation of one of the minor activities that are present in the website for lessons in this minor activity we can flip a coin which ellaborates the concept of 50% of chance of getting heads and 50% chance of getting tails when you flip a coint. There are some of the really interesting major activities in lessons we will discuss them and refer to one of them in next figure.

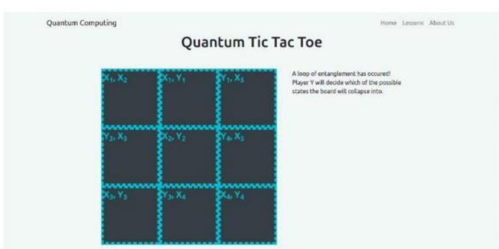


Fig 8. Major Activity

In Fig 8. It is the visual representation of one of Major activites that are present in each lessons. This activity is called Quantum tic-tac-toe is a "quantum generalization" of tic-tac-toe in which the players' moves are "superpositions" of plays in the classical game. The game was invented by Allan Goff of Novatia Labs, who describes it as "a way of introducing quantum physics without mathematics", and offering "a conceptual foundation for understanding the meaning of quantum mechanics in this game instead of having single move at a time for each player like traditional tic-tac-toe game, in quantum tic-tac-toe each player have two moves which describes the concept superposition.

C. *Algorithm of the System:*

Step 1: Start : With the start of the website show the Landing Page to the User Fig 2. where he can choose the lesson, activity or can give feedback or bug report.
Step 2: Making Choice of Lesson: After choosing the lesson the user will be redirected to the lesson content page of that corresponding lesson which can be visualise by Fig 6. And in the bottom of the psge he can go to the Major Activity of that corresponding lesson.
Step 3: Choosing Activity : User can also choose the activity of his choice from landing page as demonstrated in Fig 5. And then user will redirected to the activity page their choice.
Step 4: Interaction with Activity: When user will be redirected to the Activity page there they can interact with the activity and get to know the and understand the concept associated with that. And interactive activities will make their learning enthusiasm alive.

D. Implementation Stage:

Implementation is the phase of implementing and it will help in putting all the planned activities into action and moving the project to service provision. Languages used to implement Quantum Learning website system are as follows:

1. HTML: Hypertext Markup Language, the basic function is creating web pages. The goal of the web browser is to read the documents as web-pages; and it is also possible to include scripts written in several languages, such as JavaScript, which an impact on the behavior of web pages[3]
2. CSS: Cascading Style Sheet, is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. Using this we have decorate this web-pages with awesome colors tables and bars.[4]
3. JavaScript: A programming language developed for the design of interactive sites and creating web applications. JavaScript can interact effectively with HTML source code, enabling web authors access to their sites with dynamic content.[5]
4. React: (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.[6]
5. Bootstrap: is a free and open-source CSS

framework directed at responsive, mobile-first and frontend web development. It contains CSS- and (optionally) JavaScript based design templates for typography, forms, buttons, navigation, and other interface components. Bootstrap is the seventh-most-starred project on GitHub, with more than 142,000 stars, behind free-code-camp (almost 312,000 stars) and marginally behind Vue.js framework[7]

Programs used to implement ingredients to recipe are as follows:

1. Visual Studio Code (VS code) : is a free source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.[8]
2. Google Chrome : Google Chrome is a cross-platform web browser developed by Google. It was first released in 2008 for Microsoft Windows, and was later ported to Linux, macOS, iOS, and Android where it is the default browser built into the OS. The browser is also the main component of Chrome OS, where it serves as the platform for web applications.[9]

V. ADVANTAGES OF SYSTEM

Advantages of this system will be to the use of technology for making the life easier of those who want to learn quantum computing or those who like learning with the help of activities which made them more interactive and easy to understand. This Website will provide quality content relevant to the topic and also includes interactive actvities and. This Website will do all the Time Consuming Task of searching contents here and there and finding activities in just few clicks. This Website is having efficient, fast and easy to use interface.

VI. FUTURE ENHANCEMENT:

In future enhancement we are planning to make this project as an open source project and let other developers and people who are learning quanutm computing and want to contribute in this field to share their knowledge by adding their content to the website and make it more descriptive day by day.

VII. COCLUSION

The system eases the access the quality content and interactive activities by providing them in a single website and minimise the hasle of finding content to the different webistes and books to understand the concepts of Quantum Computing.

AUTHORS

First Author - Siddharth Raj
(rajsiddharth21@gmail.com)
Second Author – Prerna Sharma
Third Author – Dr. Jaykumar V.

REFERENCES

[1] Quantum Atlas: https://quantumatlas.umd.edu/
[2] "Quantum Computing for the Quantum Curious" by Ciaran Hughes, Joshua Isaacson, Anastasia Perry, Ranbel F. Sun, Jessica Turner.
[3] HTML : https://en.wikipedia.org/wiki/HTML
[4] CSS : https://en.wikipedia.org/wiki/CSS
[5] JavaScript : https://www.javascript.com/
[6] React:
https://en.wikipedia.org/wiki/React_(JavaScript_library)
[7] Bootstrap : https://getbootstrap.com/
[8] Visual Studio Code : https://code.visualstudio.com/
[9]GoogleChrome:
https://en.wikipedia.org/wiki/Google_Chrom

CHAPTER 7
RESULT AND CONCLUSION

Result and Advantages of this system will be to the use of technology for making the life easier of those who want to learn quantum computing or those who like learning with the help of activities which made them more interactive and easier to understand. This Website will provide quality content relevant to the topic and also includes interactive activities and. This Website will do all the Time-Consuming Task of searching contents here and there and finding activities in just few clicks. This Website is having efficient, fast and easy to use interface.

7.1 FUTURE SCOPE

In future enhancement we are planning to make this project as an open-source project and let other developers and people who are learning quantum computing and want to contribute in this field to share their knowledge by adding their content to the website and make it more descriptive day by day

7.2 CONCLUSION

The system eases the access the quality content and interactive activities by providing them in a single website and minimize the hassle of finding content to the different websites and books to understand the concepts of Quantum Computing.

REFERENCES

[1]HTML: https://en.wikipedia.org/wiki/HTML

[2] CSS: https://en.wikipedia.org/wiki/CSS

[3] JavaScript: https://www.javascript.com/

[4] React: https://en.wikipedia.org/wiki/React_(JavaScript_library)

[5] Bootstrap: https://getbootstrap.com/

[6] Visual Studio Code : https://code.visualstudio.com/

[7] GoogleChrome: https://en.wikipedia.org/wiki/Google_Chrom

[8] Quantum Atlas: https://quantumatlas.umd.edu/

[9] “Quantum Computing for the Quantum Curious” by Ciaran Hughes, Joshua Isaacson, Anastasia Perry, Ranbel F. Sun, Jessica Turner

[10] Generalized Quantum Tic-Tac-Toe Ananya Kumar1, Ang Yan Sheng1, and Chai Ming Huang∗1 1NUS High School, 20 Clementi Avenue 1, Singapore 129953

[11] https://html.com/

[12] https://www.w3.org/Style/CSS/Overview.en.html

[13] https://www.javascript.com/

[14] https://reactjs.org/