**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**

RV COLLEGE OF ENGINEERING **®**

BENGALURU-560059

(Autonomous Institution Affiliated to VTU, Belagavi)

**UG 4th semester DAA Project Synopsis**

# For

“ **Simulation on”**

**“Prims Algorithm – Finding minimum Cost Spanning Tree”**

**Course: Design and Analysis of Algorithms (CD343AI)**

# Submitted By

1. **Ishan Guptha 1RV22IS023**
2. **Prajwal Patgar 1RV22IS044**
3. **Shubham Upadhyay 1RV22IS063**
4. **Yogish HG 1RV23IS407**

# Under the Guidance of

Swetha S

Assistant professor

Department of ISE, RVCE

**Abstract**

This project focuses on the simulation of Prim's algorithm, a greedy algorithm used to find the Minimum Cost Spanning Tree (MCST) of a connected, undirected, and weighted graph. The MCST is crucial in network design, such as in telecommunication networks, transportation networks, and circuit design, where the goal is to connect all nodes (vertices) with the minimum total edge weight.

The project mainly aimed to implement Prim's algorithm to demonstrate how the algorithm selects edges sequentially, ensuring that at each step, the smallest edge that connects a new vertex to the growing tree is chosen. The simulation provides a visual representation of the graph and the step-by-step process of constructing the MCST, helping users understand the underlying principles and decisions made by the algorithm.

The project also includes an interactive interface where users can input their own graph data, visualize the progression of the algorithm, and observe the formation of the spanning tree in real-time. By simulating Prim's algorithm, this project aims to enhance understanding of algorithmic strategies for solving optimization problems in graph theory and to illustrate the practical applications of these concepts in real-world scenarios.

1. **Requirements Specification:**
   1. **Hardware Requirements**

*     **Processor**: Intel Core i5 or AMD equivalent, or better.

*     RAM: 8 GB or higher.

*     **Storage:** At least 256 GB SSD (Solid State Drive) for fast read/write operations.

*     **Graphics**: Integrated graphics card is sufficient, but a dedicated graphics card (e.g., NVIDIA GeForce GTX series) is recommended for enhanced visualization performance.

*     **Operating System:** Windows 10/11, macOS, or a Linux distribution (e.g., Ubuntu).

* **Uninterruptible Power Supply (UPS):** To prevent data loss in case of power outages, especially during long simulation sessions
  1. **Software Requirement’s**
* **Operating System:**
* Windows 10/11 (64-bit)
* macOS 10.15 or later
* Linux (Ubuntu 20.04 or later)
* **Programming Language:**
  + HTML5: For structuring the web page content.
  + CSS3: For styling and layout of the web page.
  + JavaScript (ES6+): For scripting and implementing the logic using React.js.
* **Development Environment:**
* npm or Yarn: For package management.
* React.js: For building the interactive frontend interface.
* Visual Studio Code or Atom: Preferred code editors for development, with extensions for JavaScript/React.
* **Libraries and Frameworks:**
* React.js: For creating the user interface and managing the component-based structure of the application.
* D3.js or Cytoscape.js: For graph visualization, to render and animate the graph and the algorithm’s steps.
* Bootstrap or Material-UI: For UI styling and responsiveness.
* **Version Control:**
* Git: For version control and collaboration.
* GitHub or GitLab: For hosting the project repository.
  1. **User Requirements:**

1. **User-Friendly Dashboard:**

* The application must include a user-friendly dashboard that facilitates easy navigation through the features related to Prim's algorithm.

1. **Interactive Quiz:**

* The application must provide an interactive quiz to test users' understanding of Prim's algorithm and reinforce their learning.

1. **Comprehensive Documentation:**

* The application must offer detailed documentation, including theoretical background, step-by-step explanations, pseudocode, and practical applications of Prim's algorithm.

1. **Improved User Interface:**

* The application must feature an intuitive and visually appealing user interface that supports easy navigation and engaging interactions.

1. **Real-Time Visualization of Prim's Algorithm:**

* The application must visualize the step-by-step process of Prim's algorithm in real-time, allowing users to see how the minimum spanning tree is constructed.

1. **Customizable Graph Visualization:**

* Users must be able to customize the graph by setting various vertices and edges, enabling the exploration of different scenarios and graph structures.

1. **Minimum Cost Spanning Tree with Total Cost:**

* The application must provide the minimum cost spanning tree and its total cost for any given graph, demonstrating the algorithm's efficiency in minimizing connection costs.

1. **Contact for Clarification:**

* A Contact page must be available for users to reach out for clarification or additional support regarding Prim's algorithm.

1. **Continuous Updates and Improvements:**

* The application must be updated regularly with new features and improvements to enhance the user experience and keep the content relevant.

1. **Downloadable Resources:**

* The application must provide access to downloadable resources such as lecture notes, slides, and reference materials to support users' learning journeys.

1. **Objectives**

The primary objective of this project is to develop an interactive and educational simulation of Prim's algorithm, designed to help users understand and apply this fundamental graph theory algorithm effectively. The project aims to provide a user-friendly dashboard that facilitates easy navigation through various features related to Prim's algorithm. By integrating real-time visualization, the project enables users to observe the step-by-step construction of the Minimum Cost Spanning Tree (MCST), enhancing their comprehension of the algorithm’s functionality. Additionally, the project includes interactive quizzes to test and reinforce users' understanding, comprehensive documentation for in-depth learning, and a customizable graph visualization tool that allows exploration of different graph scenarios.

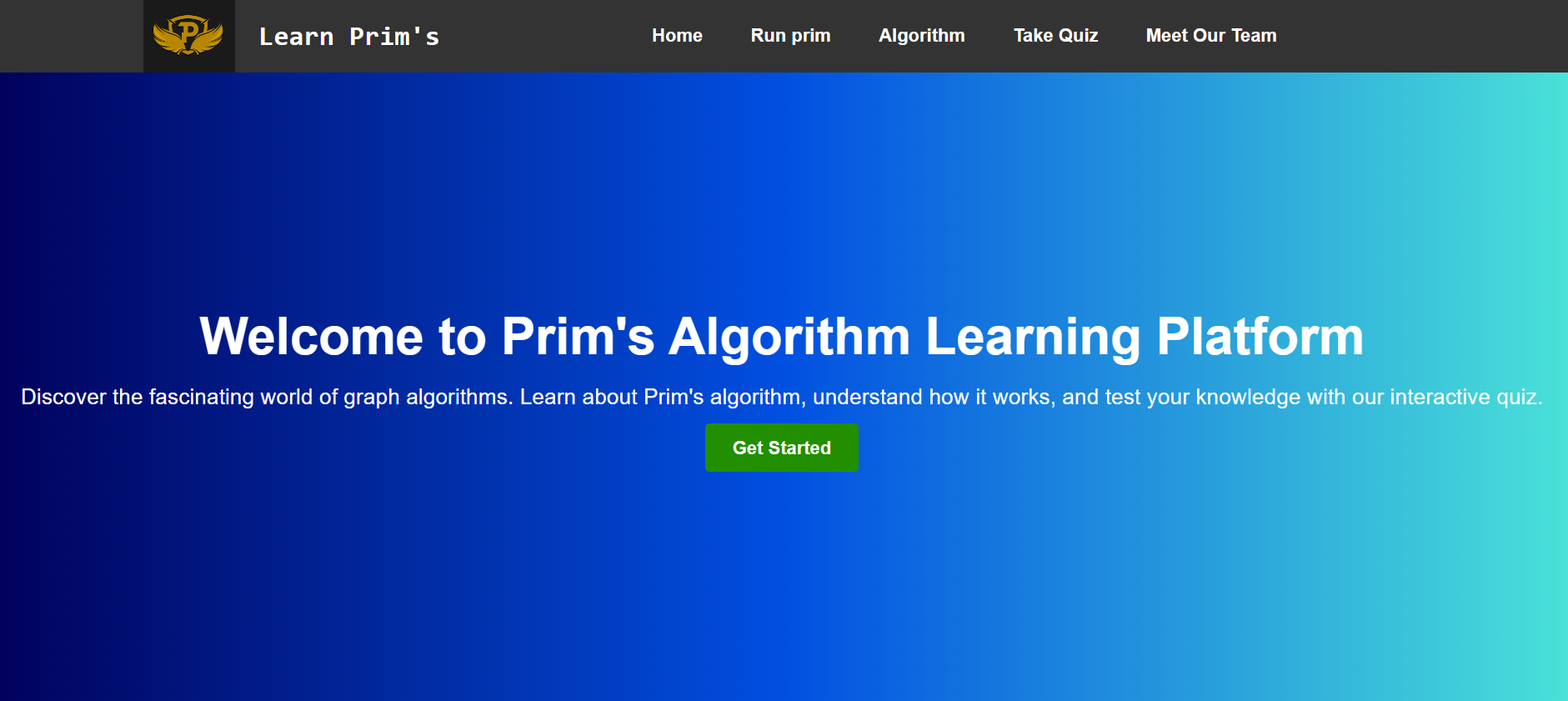
To further support users, the project offers accessible contact options for clarification, downloadable educational resources, and a certification system to validate proficiency. Through continuous updates and improvements, the project strives to deliver an engaging and educational experience that bridges theoretical concepts with practical application, ensuring users gain valuable insights into the efficiency and utility of Prim's algorithm.

1. **Methodology**

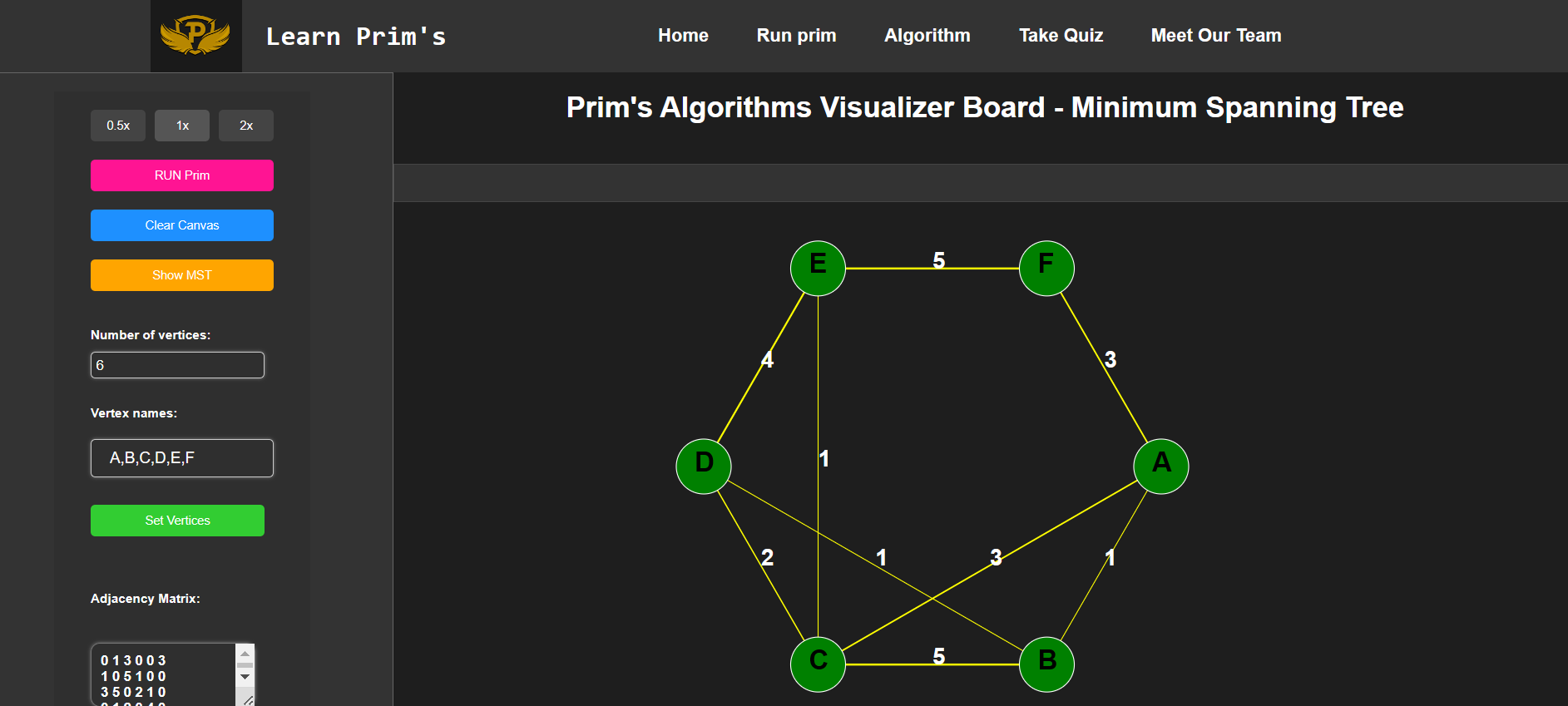
The simulation begins with an initial set of vertices and edges defined by the user, representing the weighted graph. Prim's algorithm is then applied to this graph, starting from an arbitrary vertex. The algorithm proceeds by repeatedly selecting the smallest edge that connects a new vertex to the existing spanning tree. At each step, the algorithm ensures that the growing tree remains connected and that no cycles are formed.

To facilitate understanding, the simulation provides a real-time, visual representation of the graph and the algorithm's progression. Each vertex and edge is clearly displayed, with the selected edges being highlighted as they are added to the spanning tree. The simulation also displays the current total cost of the spanning tree at each step, allowing users to observe how the algorithm minimizes the overall connection cost.

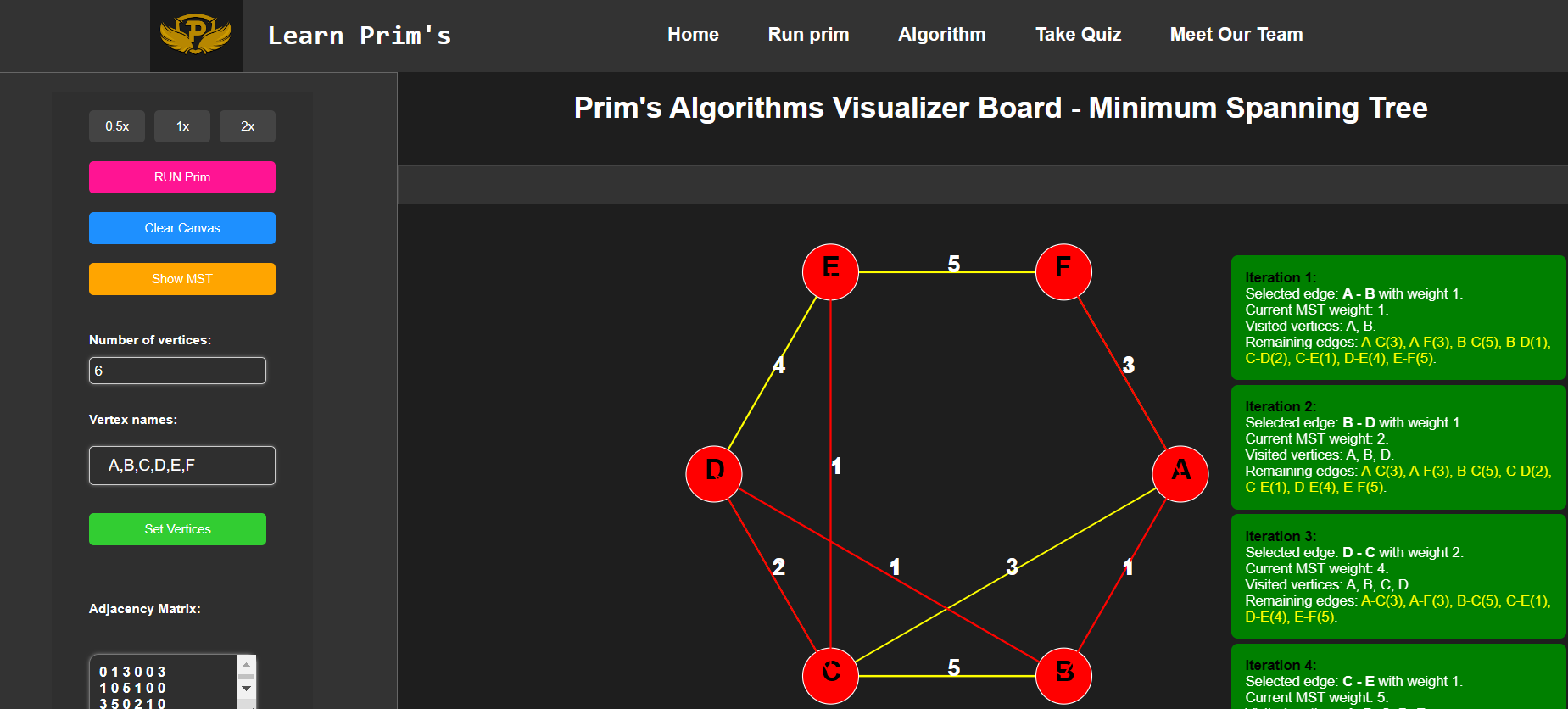
1. **Expected Outcome**

****

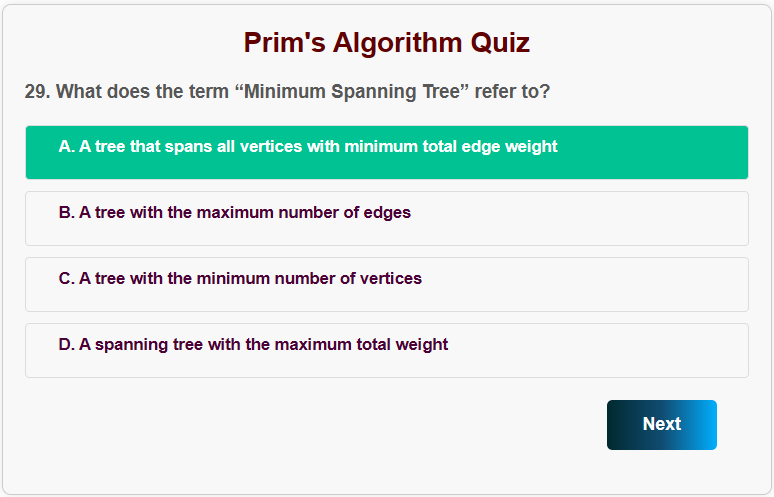
**Fig4.1:** Home Page of Our Website

****

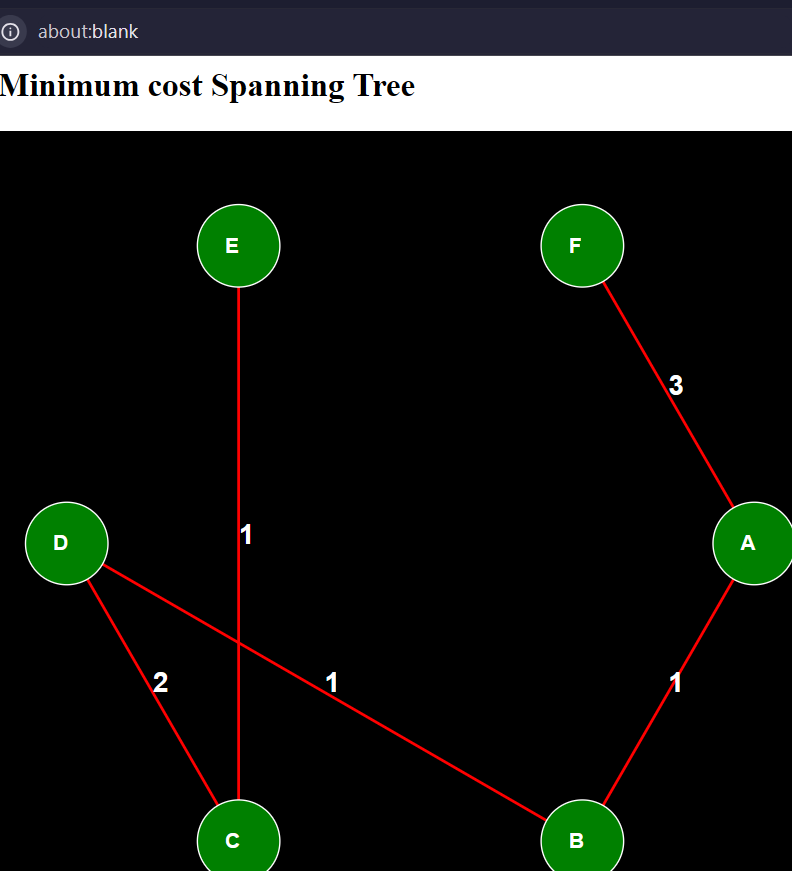
**Fig 4.2:** Primes Visualization Board

****

**Fig 4.3:** Calculating Minimum Cost Spanning Tree

****

**Fig 4.4:** Quiz Component



**Fig 4.5:** Minimum Cost Spanning Tree

1. **How to Run the Project**

**Getting Started with Create React App**

This project was bootstrapped with Create React App.

**Available Scripts**

In the project directory, you can run:

**npm start**

Runs the app in the development mode.

Open **http://localhost:3000** to view it in your browser.

The page will reload when you make changes.

You may also see any lint errors in the console.

**npm test**

Launches the test runner in the interactive watch mode.

See the section about running tests for more information.

**npm run build**

Builds the app for production to the build folder.

It correctly bundles React in production mode and optimizes the build for the best performance.

The build is minified and the filenames include the hashes.

Your app is ready to be deployed!

See the section about deployment for more information.

**npm run eject**

**Note: this is a one-way operation. Once you eject, you can't go back!**

If you aren't satisfied with the build tool and configuration choices, you can eject at any time. This command will remove the single build dependency from your project.

Instead, it will copy all the configuration files and the transitive dependencies (webpack, Babel, ESLint, etc) right into your project so you have full control over them. All of the commands except eject will still work, but they will point to the copied scripts so you can tweak them. At this point you're on your own.

You don't have to ever use eject. The curated feature set is suitable for small and middle deployments, and you shouldn't feel obligated to use this feature. However we understand that this tool wouldn't be useful if you couldn't customize it when you are ready for it.

**Learn More**

You can learn more in the Create React App documentation.

To learn React, check out the React documentation.

**Code Splitting**

This section has moved here: https://facebook.github.io/create-react-app/docs/code-splitting

**Analyzing the Bundle Size**

This section has moved here: https://facebook.github.io/create-react-app/docs/analyzing-the-bundle-size

**Making a Progressive Web App**

This section has moved here: https://facebook.github.io/create-react-app/docs/making-a-progressive-web-app

**Advanced Configuration**

This section has moved here: https://facebook.github.io/create-react-app/docs/advanced-configuration

**Deployment**

This section has moved here: https://facebook.github.io/create-react-app/docs/deployment

**npm run build fails to minify**

This section has moved here: https://facebook.github.io/create-react-app/docs/troubleshooting#npm-run-build-fails-to-minify