PROJECT REPORT: ISORT (The Sorting Visualizer)

SOFTWARE ENGINEERING

MCA 1ST YEAR

SECTION 1

PROJECT MEMBERS: RISHABH CHOPRA & AMIT KUMAR MISHRA

**Background**

Sorting algorithms are an essential aspect of computer science and play a crucial role in data processing and analysis.

The visualization of sorting algorithms is an effective way of understanding how these algorithms work. The Sorting Visualizer project aims to provide a visual representation of various sorting algorithms to enhance understanding and analysis.

This project uses web technologies, including HTML, CSS, and JavaScript, to create an interactive user interface that enables users to observe and compare sorting algorithms.

**Existing Systems**

The existing system for sorting visualizers is manual and lacks a visual representation of sorting algorithms. They are however not user-friendly as most of the functionality is hidden in sidebars, etc.

**Problem with Existing System**

The existing system for sorting visualizers is mostly limited to text-based outputs, which can be difficult to understand for individuals without programming experience. While some sorting algorithms can be visually demonstrated through diagrams and graphs, these methods are not interactive and do not offer customization options. The Sorting Visualizer project aims to fill this gap by providing a comprehensive visual representation of various sorting algorithms through a user-friendly interface with customization options.

**Objective**

1. The main objective of this project is to create a web application as a visualization tool.
2. The UI will provide various sorting algorithms to choose from whose working will be visualized.
3. It will also provide a control button to adjust the sorting speed along with a dry run.
4. The data set is represented as vertical bars with the height of their respective values. After the sorting is started, the stepwise arrangement of data in ascending order based on their value/height will be visualized in the UI.

**Process Model**

A process model for a software engineering project is chosen based on the  nature of the project and application, the methods and tools to be used, and the required controls and deliverables.

The Model used to build **"iSort"** is the **Iterative Process Model**.

The **iterative model** is a software development approach that builds a system through repeated cycles (iterations). Each iteration involves planning, designing, coding, testing, and evaluating a small part of the system. Feedback from each iteration is used to refine the system in the next cycle. This allows for gradual improvement, early problem detection, and flexibility in adapting to changing requirements.

**Key Phases in the Iterative Model:**

1. **Requirements Gathering**:
   * In the first iteration, core requirements are gathered, but not all are defined in detail at once. As iterations progress, the understanding of the system improves, and more specific requirements are identified.
2. **Design**:
   * A high-level design is created based on the initial set of requirements. As new requirements emerge in later iterations, the design is refined to accommodate additional features or address feedback.
3. **Implementation**:
   * A working version of the software is developed during each iteration. Initially, only the most essential parts of the system are implemented. As development continues, new features are added in successive iterations.
4. **Testing**:
   * Each iteration includes testing, ensuring that errors and issues are identified and corrected early. This makes it easier to fix bugs and refine functionality before the next iteration.
5. **Evaluation**:
   * After each iteration, feedback is collected from users or stakeholders. The evaluation phase helps decide what needs improvement or enhancement in the next iteration.
6. **Refinement**:
   * Based on feedback and test results, the software is modified and improved in the next iteration. The cycle repeats, with continuous enhancements until the final system is ready.

