**Capstone Project Submission**

**Instructions:**

i) Please fill in all the required information.

ii) Avoid grammatical errors.

| **Team Member’s Name, Email and Contribution (Ignore if worked individually):** |
| --- |
|  |
| **Please paste the GitHub Repo link.** |
| https://github.com/pikeshkumar/quiz-management-system.git |
| **Please paste the weblink to deployment.** |
| https://quiz-eight-pi-73.vercel.app/ |
| **Please paste the link to the video demonstration of the project.** |
| https://drive.google.com/file/d/13Modnsfg7Jk4OTIoStlQqJM4\_Z7JkdlZ/view?usp=drive\_link |

| **Please write a short summary of your Capstone project and its components.**  **Describe the problem statement, your approaches and your conclusions. (200-**  **400 words)** |
| --- |
| **Project Summary: Quiz Management System**  **The project involves developing a Quiz Management System using React, MUI (Material-UI), and local storage for data persistence. The system allows users to create, edit, play, and delete quizzes. The main components include a Quiz Creation module (CreateQuiz.js) and a Quiz Display and Management module (Table.js). The system provides a seamless user experience with a responsive and intuitive interface.**  **Key Features:**  **1. Quiz Creation:**  **- Users can create quizzes by specifying the title, description, and individual questions.**  **- Questions can be of various types: single-choice, multiple-choice, short answer, or descriptive.**  **2. Quiz Management:**  **- Existing quizzes are displayed in a table with details like quiz number, title, type, status, and creation date.**  **- Users can edit, play, or delete quizzes from the management interface.**  **3. Quiz Playback:**  **- Users can play quizzes, and the system records and displays the results.**  **- Quiz types include single-choice and multiple-choice.**  **4. Responsive Design:**  **- The application features a responsive design for optimal user experience across different devices.**  **Methodology:**  **1. Frontend Development:**  **- React and Material-UI were employed for building the frontend, ensuring a modern and visually appealing user interface.**  **- Components such as tables, buttons, and modals from Material-UI were customized for the project's requirements.**  **2. Data Persistence:**  **- Local storage is used to persistently store quiz data. Each quiz is stored with a unique key based on its title.**  **3. User Navigation:**  **- The react-router-dom library is utilized for seamless navigation between different sections of the application.**  **4. User Feedback:**  **- The application provides informative feedback to users through alerts and notifications, enhancing the overall user experience.**  **Outcome:**  **The project results in a robust Quiz Management System that empowers users to effortlessly create, manage, and engage with quizzes. The integration of React and Material-UI ensures an aesthetically pleasing and responsive user interface. The utilization of local storage facilitates data persistence, allowing users to revisit and interact with their quizzes even after closing the application.**  **Future Enhancements:**  **Future enhancements may include user authentication, more advanced quiz types, and improved analytics for quiz performance tracking.**  **The combination of cutting-edge frontend technologies and thoughtful design choices culminates in a user-friendly and efficient Quiz Management System.** |