Technologies Used:

ReactJS:
Frontend library for building user interfaces. React enables the creation of dynamic and responsive user interfaces, allowing for efficient updates and rendering.

Node.js:
Backend JavaScript runtime that executes server-side code. Node.js is used to handle server-side logic, API requests, and communicate with the database.

MongoDB:
NoSQL database used for storing data in a flexible, JSON-like format. MongoDB is suitable for handling unstructured or semi-structured data, making it a good choice for a leave management system.

Mongoose:
MongoDB object modeling for Node.js. Mongoose simplifies interactions with MongoDB by providing a schema-based solution for modeling application data.

Nodemailer:
A module for Node.js that allows the application to send emails. In the context of a leave management system, Nodemailer is likely used to notify users (e.g., employees and admins) about leave requests, approvals, or rejections.

Vercel:
A cloud platform used for hosting frontend applications. Vercel provides an easy deployment process, automatic scaling, and CDN (Content Delivery Network) capabilities for improved performance.

Design Decisions:

React for Dynamic UI:
React was chosen for the frontend to build a dynamic and responsive user interface. Its component-based architecture makes it easy to manage and reuse UI elements.

Node.js for Backend:
Node.js is employed on the server side to handle HTTP requests, manage authentication, and interact with the MongoDB database. Its non-blocking I/O model is well-suited for handling concurrent operations.

MongoDB for Data Storage:
MongoDB was chosen as the database for its flexibility in handling JSON-like documents. This is beneficial for managing leave requests, user profiles, and other related data in a scalable and adaptable manner.

Mongoose for Data Modeling:
Mongoose is used to define and enforce a schema for data stored in MongoDB. This helps maintain data integrity and facilitates easy querying and validation.

Nodemailer for Email Notifications:
Nodemailer is integrated to send email notifications. This feature is crucial for notifying employees and administrators about leave request submissions, approvals, and rejections.

Vercel for Hosting Frontend:
Vercel is chosen for hosting the frontend due to its ease of use, seamless integration with Git, automatic deployments, and global CDN, ensuring fast and reliable access to the application.

User Authentication:
User authentication is implemented to secure the application. Employees and admins need to log in to access their respective dashboards. This enhances data privacy and ensures that only authorized individuals can perform actions.

Role-Based Access Control (RBAC):
RBAC is likely implemented to distinguish between regular employees and administrators. Admins have the authority to approve or reject leave requests, while employees can submit requests and view their leave history.

Dashboard for User Interaction:
Separate dashboards for employees and admins provide a user-friendly interface for submitting, tracking, and managing leave requests. Features such as displaying leave history and real-time updates contribute to a positive user experience.

Secure Communication:
Security measures, such as HTTPS, are likely implemented to ensure secure communication between the frontend and backend, protecting sensitive user information during data transmission.

By combining these technologies and making thoughtful design decisions, the leave management site is equipped with the necessary features for effective leave request management, user authentication, and communication.