TrustGate

Controlled Digital Asset Release System

Team member's roll number, Name

Member-1: 2501118 Rohan Sandeep Mane

Member-2: 2501197 Anuja Chandrakant Wadnere

User Requirements

Functional Requirements

- User registration and secure login
- Upload and store important digital documents
- Add and manage trusted contacts
- Configure inactivity duration
- Automatic detection of user inactivity
- Controlled release of selected data to trusted contacts
- Secure notification through email

Non-Functional Requirements

- Data security and confidentiality
- High availability using cloud infrastructure
- Scalability to support multiple users
- User-friendly interface
- Reliable cloud storage and backup

Module Specifications

1. User Authentication Module

- User signup, login, and logout
- Secure authentication using cloud-based identity services

2. Digital Asset Management Module

- Upload and manage digital files
- Secure storage of files in cloud storage
- Maintain file metadata for tracking and access control

3. Trusted Contact Management Module

- Add and update trusted contacts

- Assign controlled access permissions

4. Inactivity Monitoring Module

- Track last user activity timestamp
- Detect inactivity based on user-defined duration

5. Controlled Data Release Module

- Verify inactivity condition before data release
- Notify trusted contacts through secure email alerts
- Provide limited and secure access to selected data

6. Security & Access Control Module

- Role-based access control (RBAC)
- Protection against unauthorized data access

Technology to be Used

Frontend

- **React.js** – for building a scalable and interactive user interface
- **HTML5** – for structuring web pages
- **CSS3** – for responsive and clean UI design
- **JavaScript (ES6+)** – for client-side logic and interactions

Backend / API Layer

- **Node.js** – server-side runtime environment
- **Express.js** – framework for building RESTful APIs
- **REST API Architecture** – for communication between frontend and backend

Database

- **MongoDB (MongoDB Atlas – Cloud Database)**
  - Stores user profiles, trusted contacts, activity logs, and file metadata
  - Selected for schema flexibility and cloud-native scalability

Cloud & Storage Services

- **Firebase Cloud Storage** – secure cloud storage for digital documents
- **Firebase Authentication** – user identity management and access control

Authentication & Security

- **JWT (JSON Web Tokens)** – secure API authentication
- **Role-Based Access Control (RBAC)** – controlled access for users and trusted contacts

- **Password Hashing (bcrypt)** – secure credential storage

Automation & Scheduling

- **Node-Cron / Firebase Scheduled Functions** – for inactivity monitoring and automated event triggers

DevOps & Deployment

- **Git & GitHub** – version control and team collaboration
- **Docker** – containerization of backend services
- **GitHub Actions** – CI/CD pipeline automation

Development Tools

- **VS Code** – source code editor
- **Postman** – API testing and validation
- **MongoDB Compass** – database management

Conclusion

This project aims to provide a secure and cloud-based solution for managing digital assets and ensuring controlled access during prolonged user inactivity. The system demonstrates practical implementation of cloud computing concepts, security mechanisms, automation, and scalable system design.