**COVERWALL**

DATABASE ARCHITECTURE OVERVIEW

**DATABASE REQUIREMENTS FOR COVERWALL CENTRAL SERVER IMPLEMENTATION:**

**1. Database System:**

* **Type: Relational Database Management System (RDBMS) given the structured nature of the data and relationships.**
* **Suggested Systems: MongoDB.**
* **Version: Latest stable version to ensure security and feature availability.**

**2. Schema Design:**

* **Tables for Schools, Teachers, Students, Videos, Policies, ProxyLogs, AgentConfig, AI\_Models, AI\_Insights, AI\_TrainingData, AI\_Logs, Classes, RiskEvents, and any other identified entities. (See Below)**
* **Normalized schema to reduce data redundancy and improve data integrity.**

**3. Indexing:**

* **Indexes on frequently queried columns such as school\_id, teacher\_id, student\_id, video\_id, etc.**
* **Consider composite indexes for columns that are often queried together.**

**4. Relationships & Constraints:**

* **Clearly defined foreign key relationships between tables.**
* **Unique constraints on columns like email addresses.**
* **Check constraints where necessary, e.g., risk scores between 0 and 100.**

**5. Security:**

* **Secure authentication mechanism for database access.**
* **Data encryption at rest and in transit.**

**6. Backup & Recovery:**

* **Regularly scheduled backups with both full and incremental options.**
* **Secure storage of backup data, possibly off-site or in a cloud environment.**
* **Clearly defined disaster recovery procedures.**

**7. Performance & Scalability:**

* **Database optimization techniques, including query optimization.**
* **Potential for horizontal scaling (adding more servers) or vertical scaling (adding more resources to the existing server).**
* **Caching mechanisms for frequently accessed data.**

**8. Data Retention & Archiving:**

* **Policies defining the duration of data retention.**
* **Mechanisms for archiving old data without deletion.**

**9. Integration Points:**

* **API endpoints for interaction with other systems, especially the desktop agent and the school administration server.**
* **Secure methods for data exchange, possibly using JSON or XML formats.**

**10. Data Migration & Seeding:**

* **Tools or scripts for initial data population (seeding).**
* **Procedures for migrating existing data, if applicable.**

**11. Development & Testing Environment:**

* **Separate database instances for development, testing, and production.**
* **Data anonymization tools for testing environments to ensure privacy.**

**12. Monitoring & Logging:**

* **Tools to monitor database health, performance, and usage.**
* **Logging mechanisms to track database events, especially critical or unusual activities.**

**13. Documentation:**

* **Comprehensive documentation detailing the database schema, relationships, API endpoints, and any custom functions or procedures.**
* **Regular updates to the documentation as the system evolves.**

DATABASE SCHEMA & LAYOUT

**LAYOUT**

**Coverwall School Administration Server Database:**

**1. Users:**

* UserID, Name, Email, Password (hashed), Role (District Admin, School Admin, Teacher, Student), AssociatedSchoolID, AssociatedTeacherID (for students).

**2. Schools:**

* SchoolID, SchoolName, DistrictID, Address, SubscriptionStatus, SubscriptionEndDate.

**3. Districts:**

* DistrictID, DistrictName, AdminID.

**4. Videos:**

* VideoID, YouTubeLink, EmbedCode, UploadedBy (TeacherID), ApprovalStatus, AssociatedClassID.

**5. Classes (if teachers' classes are identified):**

* ClassID, ClassName, TeacherID, SchoolID.

**6. Content Policies:**

* PolicyID, PolicyName, PolicyDetails, CreatedBy (Admin/Teacher ID), SchoolID.

**7. Billing & Sales:**

* TransactionID, SchoolID, Amount, TransactionDate, SubscriptionType, PaymentGatewayReference.

**8. Feedback:**

* FeedbackID, UserID, FeedbackText, Timestamp.

**9. Notifications & Alerts:**

* NotificationID, UserID, NotificationText, Status, Timestamp.

**Coverwall Central Server Database:**

**1. Proxy Requests:**

* RequestID, SourceIP, RequestedURL, Timestamp, ResponseStatus (Allowed/Blocked), SchoolID.

**2. Master Policies:**

* MasterPolicyID, PolicyName, PolicyDetails.

**3. AI/ML Insights:**

* InsightID, AssociatedVideoID, InsightDetails, RiskScore, Timestamp.

**4. Software Agent:**

* AgentID, Version, ConfigurationDetails, LastUpdateTimestamp.

**5. System Health & Metrics:**

* MetricID, MetricName, Value, Timestamp.

**6. Security Logs:**

* LogID, EventDetails, SourceIP, Timestamp, ActionTaken.

**7. AI/ML Models:**

* ModelID, ModelName, Version, LastTrainedTimestamp.

**8. Backup & Recovery:**

* BackupID, BackupTimestamp, BackupLocation, RecoveryStatus.

**9. Opt-out Requests (if applicable):**

* RequestID, UserID, OptOutDetails, Timestamp.

**10. System Logs & Monitoring:**

* LogID, EventDetails, Timestamp.

**SCHEMA**

**1. Schools Table:**

* **school\_id**: Unique identifier for each school.
* **school\_name**: Name of the school.
* **address**: Physical address of the school.
* **subscription\_status**: Indicates if the school has subscribed to the Proxy feature set (e.g., 'active', 'inactive').
* **subscription\_date**: Date when the school subscribed.
* **policy\_id**: Foreign key to the Policies table.

**2. Teachers Table:**

* **teacher\_id**: Unique identifier for each teacher.
* **school\_id**: Foreign key to the schools table.
* **class\_id**: Foreign key to the Classes table. This can be a one-to-many relationship if a teacher handles multiple classes.
* **first\_name**: First name of the teacher.
* **last\_name**: Last name of the teacher.
* **email**: Email address of the teacher.
* **password**: Hashed password for authentication.

**3. Students Table:**

* **student\_id**: Unique identifier for each student.
* **school\_id**: Foreign key to the Schools table.
* **first\_name**: First name of the student.
* **last\_name**: Last name of the student.
* **grade**: Grade or class of the student.
* **risk\_score**: A numerical value representing the student's current risk score.
* **last\_updated**: Date and time when the risk score was last updated.

**4. Videos Table:**

* **video\_id**: Unique identifier for each video.
* **teacher\_id**: Foreign key to the Teachers table.
* **url**: URL of the YouTube video.
* **title**: Title of the video.
* **description**: Description of the video.
* **status**: Approval status (e.g., 'approved', 'pending', 'denied').

**5. Policies Table:**

* **policy\_id**: Unique identifier for each policy.
* **school\_id**: Foreign key to the school's table.
* **policy\_name**: Name of the policy.
* **whitelist**: List of allowed URLs or domains.
* **blacklist**: List of blocked URLs or domains.
* **content\_filters**: List of content types or keywords to filter.

**6. ProxyLogs Table:**

* **log\_id**: Unique identifier for each log entry.
* **school\_id**: Foreign key to the school's table.
* **request\_url**: The URL that was requested.
* **request\_date**: Date and time of the request.
* **decision**: Decision made by the proxy (e.g., 'allowed', 'blocked').
* **reason**: Reason for the decision (e.g., 'blacklisted', 'content filter').

**7. AgentConfig Table:**

* **config\_id**: Unique identifier for each configuration.
* **school\_id**: Foreign key to the school's table.
* **agent\_version**: Version of the desktop agent.
* **update\_url**: URL for agent updates.
* **proxy\_endpoint**: Endpoint URL of the central server for proxy requests.

**Classes Table:**

* **class\_id**: Unique identifier for each class.
* **teacher\_id**: Foreign key to the Teachers table.
* **class\_name**: Name of the class (e.g., "Algebra I", "Biology 101").
* **subject**: Subject of the class (e.g., "Math", "Biology").
* **grade**: Grade or level of the class (e.g., "9th Grade", "10th Grade").

**RiskEvents Table:**

* **event\_id**: Unique identifier for each risk event.
* **student\_id**: Foreign key to the student’s table.
* **timestamp**: Date and time of the event.
* **event\_type**: Type of risk event (e.g., "inappropriate content access", "frequent late-night browsing").
* **event\_description**: Detailed description of the event.
* **event\_weight**: A numerical value indicating the event's severity, which contributes to the overall risk score.

**FUTURE USE: COMING SOON**

**1. AI\_Models Table:**

* **model\_id**: Unique identifier for each AI/ML model.
* **model\_name**: Name or description of the model (e.g., "Content Categorization v1.0").
* **model\_type**: Type of the model (e.g., "classification", "regression").
* **model\_version**: Version of the model.
* **last\_trained**: Date when the model was last trained or updated.
* **accuracy**: Accuracy or performance metric of the model.
* **model\_path**: Path or location where the trained model is stored.

**2. AI\_Insights Table:**

* **insight\_id**: Unique identifier for each insight generated.
* **model\_id**: Foreign key to the AI\_Models table, indicating which model generated this insight.
* **video\_id**: Foreign key to the Videos table, if the insight is related to a specific video.
* **timestamp**: Date and time when the insight was generated.
* **insight\_type**: Type of insight (e.g., "content\_warning", "sentiment\_analysis").
* **insight\_value**: Detailed value or result of the insight (e.g., "potentially inappropriate content detected").

**3. AI\_TrainingData Table:**

* **data\_id**: Unique identifier for each training data entry.
* **model\_id**: Foreign key to the AI\_Models table, indicating which model this data is associated with.
* **data\_path**: Path or location where the training data is stored.
* **data\_description**: Description or notes about the training data.
* **last\_used**: Date when this training data was last used to train a model.

**4. AI\_Logs Table:**

* **log\_id**: Unique identifier for each log entry related to AI/ML operations.
* **model\_id**: Foreign key to the AI\_Models table.
* **timestamp**: Date and time of the log entry.
* **log\_type**: Type of log (e.g., "training", "prediction", "error").
* **log\_message**: Detailed log message or description.