**Key Components**

1. **Framework and Libraries**:
   * **Flask**: Web framework for building the API.
   * **Flask-RESTX**: Provides Swagger UI and API documentation.
   * **Flask-SQLAlchemy**: ORM for MySQL database interactions.
   * **Flask-JWT-Extended**: JWT-based authentication.
   * **Flask-CORS**: Handles Cross-Origin Resource Sharing.
   * **APScheduler**: Background scheduler for periodic health checks.
   * **MySQL**: Database for storing users, services, risks, and audit logs.
2. **Database Models**:
   * **User**: Stores user credentials and roles (user, admin, etc.).
   * **Service**: Represents a service with attributes like name, description, and relationships to BIA, Status, Downtime, Integration, and Risk.
   * **BIA**: Business Impact Analysis with criticality, RTO, RPO, and dependencies.
   * **Status**: Tracks service health status (e.g., Healthy, Down).
   * **Downtime**: Logs service downtime events.
   * **Integration**: Stores third-party integration details (e.g., Slack, AWS).
   * **Risk**: Tracks risk scores and levels for services.
   * **AuditLog**: Logs actions for auditing (e.g., user signup, service creation).
3. **API Endpoints**:
   * **Auth**: /api/auth/signup, /api/auth/login for user registration and login.
   * **Services**: CRUD operations for services (/api/services), status updates (/api/services/<id>/status), BIA management (/api/services/<id>/bia), downtime logging (/api/services/<id>/downtime), and integrations (/api/services/integrations).
   * **Risk**: Risk retrieval (/api/risk/<id>), automated risk saving (/api/risk/<id>/save), manual risk input (/api/risk/<id>/manual), and risk updates (/api/risk/<id>/update).
   * **Audit**: Audit log retrieval (/api/audit).
   * **Health**: Service health check (/api/services/<id>/health).
4. **Security**:
   * JWT authentication with role-based access control (role\_required decorator).
   * Password hashing using werkzeug.security.
   * Audit logging for tracking actions.
5. **Risk Analysis**:
   * The calculate\_risk\_score function computes a risk score based on service status, BIA attributes (e.g., RTO), and dependencies.
   * Scores determine risk levels (Low, Medium, High).
6. **Health Checks**:
   * A background scheduler (run\_health\_checks) updates service statuses every minute based on the last update time.
   * Services not updated within 5 minutes are marked as Down.