Safe Report System – Interview Q&A; Document

Tech Stack Overview
Backend: Django REST Framework, Python, SQLite
Frontend: Flutter (Dart)
APIs: JWT Authentication (SimpleJWT), REST APIs
Encryption: Fernet (symmetric AES encryption)
NLP: TextBlob (sentiment analysis)
Hosting (future-ready): AWS / Render / Railway
Database Migration Ready: SQLite → PostgreSQL

1■■ Security & Encryption

Q: How did you handle key management for Fernet?
A: The encryption key is stored securely using environment variables and never hardcoded. Django's settings module references the key dynamically at runtime for enhanced security.

Q: How would you rotate encryption keys if needed?
A: Key rotation can be achieved by generating a new key and re-encrypting sensitive data, while maintaining backward compatibility during transition periods.

Q: How do you ensure encryption doesn't leak PII through logs?
A: PII is never logged in plaintext. Logging is restricted to metadata only, and debug logs are disabled in production environments.

2■■ Authentication

Q: What authentication did you use for API calls?
A: JWT (JSON Web Token) authentication implemented via Django REST Framework SimpleJWT for stateless and secure API access.

Q: Why did you choose token-based auth?
A: Token-based authentication allows better scalability for mobile and web apps, avoiding session management complexities.

Q: How do you refresh or revoke tokens?
A: Access tokens are short-lived and refresh tokens are provided for renewal. Token revocation is handled by invalidating refresh tokens in the backend.

3■■ Scaling & Deployment

Q: How would you scale this if 10,000 students used it daily?
A: By using a PostgreSQL database, caching (Redis), load balancing with Nginx, and deploying on AWS EC2 or Render with Docker containers.

Q: Where would you host it?
A: A cost-effective choice would be AWS or Render for automatic scaling and database management.

Q: How would you move from SQLite to Postgres in production?
A: By exporting data using Django's dumpdata/loaddata or a migration tool like pgloader, updating DATABASES config in settings.py, and running migrations.

4■■ Sentiment Analysis

Q: Why did you pick TextBlob vs other NLP tools?
A: TextBlob offers simplicity and sufficient accuracy for English sentiment detection without needing heavy NLP models.

Q: How would you handle multilingual reports?
A: By integrating translation APIs like Google Translate before performing sentiment analysis.

Q: Do you think the sentiment is accurate for all cases?
A: No NLP tool is perfect. TextBlob can misinterpret sarcasm or mixed emotions, so results are used as supportive data, not final judgment.

5■■ Threat Modeling

Q: What are possible attack vectors on your reporting system?
A: Spam submissions, brute-force login attempts, and XSS/SQL injections are main threats. Django ORM and authentication prevent most of these.

Q: How do you prevent an attacker from submitting spam reports?
A: By using rate-limiting middleware, CAPTCHA, and validating request tokens.

Q: What if your encryption key gets leaked?
A: Immediately rotate the key, re-encrypt data, and invalidate all tokens while investigating breach logs.

6■■ Real-World Use

Q: How would you ensure only valid institution members sign up?
A: By validating email domains or integrating with the college's existing authentication (LDAP or SSO).

Q: How could you integrate this with an existing college portal?
A: By exposing secure REST APIs or OAuth2-based integration for single sign-on.

Q: How would you handle a request for a report to be deleted?
A: Follow privacy compliance by verifying the requester's ownership before securely deleting or anonymizing the record.

7■■ Reviewer/Admin Flow

Q: Why did you use Django admin?
A: It allows rapid administrative control and easy report review without needing to build a custom dashboard initially.

Q: Any plan to build a custom dashboard?
A: Yes, future plans include a React or Flutter Web dashboard for richer analytics and better UI control.

Q: How do reviewers change status?
A: Admin or authorized reviewers can update report statuses directly in Django admin or through API endpoints.

8■■ Future Features

Q: What would you add next if you had more time?
A: Add AI-assisted report classification, real-time chat support, and analytics dashboards for institutions.

Q: Why didn’t you implement a chatbot now?
A: Due to time constraints; it would require NLP intent detection and backend workflow integration.

Q: What unique feature would make your system stand out?
A: End-to-end encryption with anonymous yet trackable reporting ensures trust and transparency in sensitive reporting.

9■■ Testing

Q: How did you test encryption?
A: By encrypting and decrypting sample data, ensuring round-trip integrity and verifying key validity.

Q: Did you write unit tests?
A: Yes, for authentication, report creation, and encryption logic using Django’s TestCase.

Q: How did you test your API endpoints?
A: Using Postman and automated test scripts to validate authentication, response codes, and edge cases.

■ Flutter

Q: Why did you pick Flutter?
A: For cross-platform compatibility, speed, and expressive UI design for both Android and iOS.

Q: How does the Flutter app handle network failures?
A: Through try-catch blocks, custom error dialogs, and connection timeout handling using Dio or http package.

Q: How do you securely store tokens on the device?
A: Using Flutter Secure Storage plugin which encrypts tokens in platform-specific secure storage (Keychain or Keystore).