**Introduction:**

This report outlines the rationale behind selecting the tech stack and database for the event management API project. It highlights key design decisions made and addresses challenges encountered during development.

**Tech Stack:**

For this project, the chosen tech stack consists of Django, Django REST Framework (DRF), Python, and integration with external APIs. The decision to use Django was based on its robustness, security features, and DRF's support for building RESTful APIs efficiently. Python was selected as the primary programming language due to its readability and extensive libraries, which streamline web development. Integration with external APIs adds weather and distance functionalities, enhancing the API's usefulness.

**Database:**

SQLite was chosen as the database management system for its simplicity, ease of setup, and portability. Despite its lightweight nature, SQLite effectively stores event data for this project. However, it's acknowledged that SQLite may have limitations regarding performance and scalability compared to larger databases like PostgreSQL or MySQL.

**Design Decisions:**

* **RESTful Architecture**: The API follows RESTful principles to ensure clarity and predictability, enhancing usability.
* **Normalization:** Database tables are normalized to reduce redundancy and maintain data integrity.
* **External API Integration:** Integration with external APIs provides additional functionalities without reinventing the wheel.
* **Error Handling:** Adequate error handling mechanisms were implemented to provide meaningful feedback to users and ensure graceful degradation in case of failures.

**Challenges and Solutions:**

* **Handling Geographic Data:** GeoDjango extensions were utilized to efficiently manage geographic data, facilitating location-based filtering and distance calculations.
* **Integration with External APIs:** Challenges such as rate limiting and handling API errors were addressed through throttling mechanisms and retry strategies.
* **Performance Optimization:** Indexes were added to database tables to improve query performance, especially for frequently accessed fields.