# COMP3011 Technical Report: EventHub API

**Module:** COMP3011 – Web Services and Web Data
**Student:** Nathaniel Sebastian (sc232ns)
**Date:** 5th February 2026
**Word Count:** ~1,800 words

## Submission Links

| Resource | Link |
|---|---|
| GitHub Repository | github.com/NathS04/comp3011-cw1-api |
| Live API | comp3011-cw1-api.onrender.com |

## 1. Compliance Checklist

| Requirement | Status | Location |
|---|---|---|
| GitHub repository | ✅ | github.com/NathS04/comp3011-cw1-api |
| API documentation PDF | ✅ | `docs/API_DOCUMENTATION.pdf` |
| Technical report PDF | ✅ | `TECHNICAL_REPORT.pdf` |
| Presentation slides | ✅ | `docs/PRESENTATION_SLIDES.pptx` |
| GenAI logs | ✅ | `docs/GENAI_EXPORT_LOGS.pdf` |
| README.md | ✅ | Root directory |
| Deployed API | ✅ | comp3011-cw1-api.onrender.com |
| Novel data integration | ✅ | Leeds TEN XML with SHA256 provenance |
| Authentication | ✅ | JWT + PBKDF2 |
| Test suite | ✅ | **41 tests passing** |

### Marker Evidence Cross-Reference

| Requirement | Evidence | File Path |
|---|---|---|
| Tests pass (41) | `pytest -q` output | `VERIFICATION_INSTRUCTIONS.md` |

| Requirement | Evidence | File Path |
|---|---|---|
| Live deployment | `/health` → `environment: prod` | comp3011-cw1-api.onrender.com/health |
| API documentation | Full endpoint reference | `docs/API_DOCUMENTATION.pdf` |
| Security headers | 7 headers on every response | `app/core/middleware.py` |
| Rate limiting | 120/min global, 10/min login | `app/core/rate_limit.py` |
| Novel dataset | Leeds TEN XML, SHA256 hashed | `scripts/import_dataset.py` , `docs/DATASET_SOURCE.md` |
| GenAI logs | Tools, failures, dates | `docs/GENAI_EXPORT_LOGS.pdf` |
| RBAC | Admin-only `/admin/*` with 403 | `app/api/admin.py` , `app/core/auth.py` |
| ETag/304 | RFC 7232 compliance | `app/core/middleware.py` |
| Presentation | Slide outline + viva prep | `docs/PRESENTATION_OUTLINE.md` |

## 2. Reproducibility

```
git clone https://github.com/NathS04/comp3011-cw1-api.git && cd comp3011-cw1-api
python3 -m venv venv && source venv/bin/activate
pip install -r requirements.txt
export DATABASE_URL="sqlite:///./app.db" SECRET_KEY="dev-secret"
alembic upgrade head
pytest -q                    # Expected: 41 passed
uvicorn app.main:app --reload
```

## 3. Dataset Provenance

| Attribute | Value |
|---|---|
| Source | Leeds Temporary Event Notices [1] |
| Provider | Leeds City Council (Data Mill North) |
| Licence | Open Government Licence v3.0 |
| Format | XML |
| Fields Mapped | Reference_Number → ID, Premises_Name → Title |

| Attribute | Value |
|---|---|
| Provenance | SHA256 hash stored in `ImportRun` table |

# 4. Architecture

```
Client → Middleware (Rate Limit, Headers, ETag) → FastAPI → Auth → Routes → CRUD → SQLAlchemy → DB
```

**Design Principles:** - Layered architecture: thin routes, fat CRUD - Middleware handles cross-cutting concerns (rate limiting, security headers, ETag) - Dual database support via Alembic (SQLite dev, PostgreSQL prod)

# 5. Outstanding Evidence

## 5.1 Test Coverage

| Category | Count | Coverage |
|---|---|---|
| Auth | 6 | Register, login, validation |
| Events CRUD | 5 | Create, read, update, delete, pagination |
| RSVPs | 4 | Create, duplicate rejection |
| Analytics | 4 | Seasonality, trending, recommendations |
| Admin/Import | 3 | Idempotency, provenance |
| RBAC | 2 | 403 for non-admin |
| Attendees | 4 | CRUD, uniqueness |
| Health | 1 | Returns metadata |
| ETag | 3 | Generation, 304, mismatch |
| Security Headers | 3 | On 200, 404, 403 |
| Error Handling | 2 | Sanitization, request_id |
| **Total** | **41** | |

Test isolation: In-memory SQLite with `StaticPool` ; tables created/dropped per test. Runtime: <1.5s.

## 5.2 Security Header Policy

All responses include:

| Header | Value | OWASP Reference |
|---|---|---|
| `X-Request-ID` | UUID v4 | Tracing/debugging |
| `X-Content-Type-Options` | `nosniff` | MIME sniffing prevention [2] |
| `X-Frame-Options` | `DENY` | Clickjacking protection [2] |
| `Referrer-Policy` | `no-referrer` | Privacy [2] |
| `Permissions-Policy` | `geolocation=(), microphone=(), camera=()` | Feature restriction [2] |
| `Cross-Origin-Resource-Policy` | `same-site` | Side-channel protection [2] |
| `Cache-Control` | `no-store` (default) / `no-cache` (ETag) | Sensitive data protection |

## 5.3 Rate Limiting

| Scope | Limit | Rationale |
|---|---|---|
| Global | 120 req/min | Prevents abuse without impacting normal use |
| `/auth/login` | 10 req/min | Credential stuffing mitigation [3] |

429 response includes `request_id` for correlation: `{"detail":"Too Many Requests","request_id":"<uuid>"}`

## 5.4 ETag/304 Standard Compliance

Implementation follows RFC 7232 [4]:

1. GET `/events` and `/events/{id}` compute SHA256 of response body
2. Response includes `ETag: "<hash>"`
3. Client sends `If-None-Match: "<hash>"`
4. Server returns 304 Not Modified with empty body if match
5. 304 still includes `ETag`, `X-Request-ID`, and security headers

## 5.5 Provenance Hashing

Each import run stores: - SHA256 hash of source file - Timestamp, duration, row counts - Enables idempotency (duplicate hash = skip)

---

## 6. Design Trade-offs

| Decision | Alternative | Trade-off | Justification |
|---|---|---|---|
| In-memory rate limiting | Redis | Single-process only | Acceptable for free-tier Render; documented as limitation |
| ETag via body hash | DB `updated_at` column | Recalculates per request | Simpler implementation; no schema changes |
| JWT without refresh | Refresh token flow | 30-min hard limit | Simpler for coursework; users re-login |
| SQLite/ PostgreSQL dual | PostgreSQL-only | Dev overhead | Alembic abstracts dialect; faster local iteration |
| Sanitized 500 errors | Detailed error messages | Less debugging info | Security first; `request_id` enables log correlation |

## 7. Analytics Endpoints

| Endpoint | Computation | Use Case |
|---|---|---|
| `/analytics/events/seasonality` | COUNT(*) GROUP BY month | Identify peak event periods |
| `/analytics/events/trending` | `(recent_rsvps × 1.5) + (total_rsvps × 0.5)` | Surface popular events |
| `/events/recommendations` | Filter by user's past RSVP categories | Personalised discovery |

**Trade-off Rationale:** Simpler algorithms chosen for coursework scope; documented as extensible via Redis caching and ML in production.

## 8. Security Implementation

| Security Measure | Implementation | File |
|---|---|---|
| Password Hashing | PBKDF2-SHA256 | `app/core/auth.py` |
| JWT Signing | HS256, 30-min expiry | `app/core/auth.py` |
| RBAC | `is_admin` flag, 403 on `/admin/*` | `app/core/auth.py`, `app/api/admin.py` |
| Rate Limiting | 120/min global, 10/min login | `app/core/rate_limit.py` |

| Security Measure | Implementation | File |
|---|---|---|
| Request Tracing | `X-Request-ID` on all responses | `app/core/middleware.py` |
| Security Headers | 6 headers on all responses | `app/core/middleware.py` |
| ETag Caching | SHA256 body hash, 304 support | `app/core/middleware.py` |
| Error Sanitization | Generic 500 message, no stack trace | `app/core/middleware.py` |

---

# 9. Deployment

**Platform:** Render.com [5]
**Database:** Managed PostgreSQL
**Config:** `render.yaml`
**Variables:** `DATABASE_URL` , `SECRET_KEY` , `ENVIRONMENT=prod` , `ALLOWED_ORIGINS` , `RATE_LIMIT_ENABLED`

**Cold Start Note:** Free tier spins down after 15 min; first request may take 30–60s.

## 9.1 Deployment Verification

Live health check confirms production configuration:

```
curl https://comp3011-cw1-api.onrender.com/health
{
  "status": "online",
  "database": "ok",
  "version": "1.0.0",
  "environment": "prod",
  "commit": "<current HEAD>",
  "timestamp": "2026-02-09T..."
}
```

**Environment detection logic** ( `app/core/config.py` ): - If `RENDER` env var exists → always `"prod"` (cannot be overridden by `.env` ) - `load_dotenv()` is skipped entirely on Render to prevent stale config leakage - Locally: respects `ENVIRONMENT` env var, defaults to `"dev"`

**Render environment variables** (via `render.yaml` + dashboard):

| Variable | Value | Source |
|---|---|---|
| `ENVIRONMENT` | `prod` | `render.yaml` |
| `DATABASE_URL` | PostgreSQL connection string | Render managed DB |
| `SECRET_KEY` | Auto-generated | `render.yaml` |
| `RATE_LIMIT_ENABLED` | `true` | `render.yaml` |

| Variable | Value | Source |
| --- | --- | --- |
| `ALLOWED_ORIGINS` | Restricted to Render + localhost | `render.yaml` |

# 10. Version Control

**Repository:** github.com/NathS04/comp3011-cw1-api

## Commit Discipline

Development followed incremental feature-branch-style commits on `main`, with meaningful conventional-commit messages (`feat:`, `fix:`, `docs:`, `test:`, `chore:`). Each commit represents a logically atomic change verified by the test suite.

```
51c27c1 fix(test): use fixture in header tests + hardened zip exclusions
dca5aa6 docs(claude): regenerate PDFs + add marker evidence checklist
23b7054 fix(test): use fixture for etag tests
f72a36b fix(config): allow rate limit true/1 and clean zip
fa1f4e5 docs(verify): bulletproof verification steps
6d2a33d docs(slides): align narrative + viva prep
acfb22c docs(genai): final audit-ready logs
5d31b97 docs(report): finalize outstanding evidence + tradeoffs
38168e5 docs(api): finalize errors/headers/etag/flows
f9d7fe6 docs(readme): sync final features + commands
cfaca6c test(final): ensure >=39 tests and green
ef8a935 chore(devtools): add ruff/mypy/bandit/pip-audit
342473d fix(errors): sanitize all 500s, no leakage
37d5574 feat(etag): ETag + 304 for list + detail
f52d165 fix(rate): 429 includes request_id + headers
```

**Observations:** - 45+ commits total, demonstrating iterative development - Clear separation of concern: feature commits, bug fixes, documentation, and testing - No monolithic dump commits — each change is reviewable in isolation

# 11. GenAI Usage

**Tools:** Google Gemini (Antigravity), Claude, ChatGPT

**Creative Applications:** 1. Compared RSVP storage approaches (embedded vs relational) 2. Evaluated rate limiting options (in-memory vs Redis) [3] 3. Researched ETag/If-None-Match RFC 7232 compliance [4] 4. Compared middleware ordering alternatives for header injection

**Failures Caught:** - Missing `requests` dependency → ModuleNotFoundError - Placeholder test with `pass` → False coverage - Deprecated `Query(regex=...)` → Warning - Headers not applied to 429 responses → Fixed middleware flow

Full logs: docs/GENAI_EXPORT_LOGS.pdf

---

## 12. Limitations & Future Work

| Limitation | Impact | Improvement |
|---|---|---|
| No token refresh | Users re-login after 30 min | Implement refresh tokens |
| In-memory rate limiting | Resets on restart; single-worker only | Redis-backed for horizontal scaling |
| No CSP header | Reduced browser-side protection | Add Content-Security-Policy |
| Basic recommendations | Location-based scoring only | Collaborative filtering with user embeddings |
| Analytics via Python loops | O(n) over all events | SQL aggregates (GROUP BY, window functions) |

**Rate limiter justification:** Render free tier runs a single worker process, so in-memory state is not lost to multi-instance divergence. The dictionary-based approach is appropriate for the deployment target. Redis migration is documented as the production upgrade path.

---

## References


[1] Leeds City Council, "Temporary Event Notices," Data Mill North, https://datamillnorth.org/dataset/temporary-event-notices (Open Government Licence v3.0)

[2] OWASP, "HTTP Security Response Headers," https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Headers_Cheat_Sheet.html

[3] OWASP, "Rate Limiting," https://cheatsheetseries.owasp.org/cheatsheets/Denial_of_Service_Cheat_Sheet.html

[4] IETF, "RFC 7232: HTTP/1.1 Conditional Requests," https://tools.ietf.org/html/rfc7232

[5] Render, "Web Services Documentation," https://render.com/docs/web-services


---

*COMP3011 CW1, University of Leeds*