STAR Community App - Security Checklist

🔐 Authentication & Authorization

Multi-Factor Authentication (MFA)

- □ Implement SMS OTP for phone verification
- □ Add email verification for account creation
- □ Require MFA for sensitive operations (token purchases, large transactions)
- □ Implement backup authentication methods (recovery codes)

JWT Token Security

- □ Use secure JWT signing algorithms (RS256 or ES256, avoid HS256)
- □ Implement short-lived access tokens (15-30 minutes)
- □ Secure refresh token storage and rotation
- □ Add token blacklisting for logout/revocation
- □ Include proper claims validation (iss, aud, exp, iat)
- □ Implement token binding to prevent token theft

Password Security

- □ Enforce strong password requirements (12+ chars, mixed case, numbers, symbols)
- □ Use bcrypt/scrypt/argon2 for password hashing (min 12 rounds)
- □ Implement password breach checking (HaveIBeenPwned API)
- □ Add password history to prevent reuse
- □ Secure password reset flows with time-limited tokens
- □ Implement account lockout after failed attempts

Role-Based Access Control (RBAC)

- □ Validate user permissions on every API request
- □ Implement principle of least privilege
- □ Add context-aware authorization (location, time, device)
- □ Secure admin role escalation processes
- □ Audit permission changes and role assignments

🔒 Data Encryption

Encryption at Rest

- □ Enable database encryption (PostgreSQL TDE/encryption at rest)
- □ Encrypt sensitive fields in database (PII, payment data)
- □ Secure file storage encryption (AWS S3 SSE or equivalent)
- □ Encrypt configuration files and secrets
- □ Use envelope encryption for highly sensitive data

Encryption in Transit

- □ Enforce HTTPS/TLS 1.3 for all communications
- □ Implement certificate pinning for mobile apps
- □ Use secure WebSocket connections (WSS)
- □ Encrypt internal service communications
- □ Validate SSL/TLS certificates properly

Key Management

- □ Use dedicated key management service (AWS KMS, Azure Key Vault)
- □ Implement key rotation policies
- □ Separate keys by environment and purpose
- □ Secure key backup and recovery procedures
- □ Monitor key usage and access

🛡️ Input Validation & Sanitization

API Input Validation

- □ Validate all inputs against strict schemas
- □ Implement input size limits and rate limiting
- □ Sanitize user inputs to prevent XSS
- □ Use parameterized queries to prevent SQL injection
- □ Validate file uploads (type, size, content)
- □ Implement CSRF protection

Service-Specific Validation

- □ **Token Transactions**: Validate amounts, prevent negative values
- □ **Booking System**: Validate dates, prevent double-booking
- □ **User Profiles**: Sanitize profile data, validate phone numbers
- □ **File Uploads**: Scan for malware, validate document types
- □ **Community Features**: Validate project/cause data, prevent spam

🚨 Vulnerability Prevention

OWASP Top 10 Protection

- □ **Injection**: Use ORMs, parameterized queries, input validation
- □ **Broken Authentication**: Implement secure session management
- □ **Sensitive Data Exposure**: Encrypt sensitive data, secure transmission
- □ **XML External Entities (XXE)**: Disable XML external entity processing
- □ **Broken Access Control**: Implement proper authorization checks
- □ **Security Misconfiguration**: Regular security configuration reviews
- □ **Cross-Site Scripting (XSS)**: Input sanitization, CSP headers
- □ **Insecure Deserialization**: Validate serialized data, use safe formats
- □ **Vulnerable Components**: Regular dependency updates and scanning
- □ **Insufficient Logging**: Comprehensive audit logging

API Security

- □ Implement rate limiting per user/IP/endpoint
- □ Use API keys for service-to-service communication
- □ Validate API versioning and deprecation
- □ Implement request/response size limits
- □ Add API gateway security policies
- □ Monitor for API abuse patterns

Database Security

- □ Use database connection pooling with authentication
- □ Implement database user privilege separation
- □ Enable database audit logging
- □ Regular database security patches
- □ Backup encryption and secure storage
- □ Monitor for suspicious database activity

📱 Mobile App Security

App-Specific Security

- □ Implement certificate pinning
- □ Secure local data storage (iOS Keychain, Android Keystore)
- □ Add app integrity verification
- □ Implement anti-tampering measures
- □ Secure inter-app communication
- □ Add biometric authentication support

Runtime Security

- □ Implement root/jailbreak detection
- □ Add debugger detection
- □ Secure API endpoint obfuscation
- □ Implement code obfuscation
- □ Add runtime application self-protection (RASP)

💰 Financial & Token Security

Token Economy Protection

- □ Implement transaction signing and verification
- □ Add double-spend prevention mechanisms
- □ Secure escrow handling with multi-signature
- □ Implement transaction limits and fraud detection
- □ Add real-time transaction monitoring
- □ Secure token minting and burning processes

Payment Security

- □ PCI DSS compliance for payment processing
- □ Tokenize payment methods (never store card data)
- □ Implement 3D Secure for card transactions
- □ Add fraud detection algorithms
- □ Secure refund and chargeback handling
- □ Monitor for suspicious payment patterns

🔍 Monitoring & Incident Response

Security Monitoring

- □ Implement SIEM (Security Information and Event Management)
- □ Set up intrusion detection systems (IDS)
- □ Monitor for brute force attacks
- □ Track privilege escalation attempts
- □ Add behavioral analytics for user accounts
- □ Implement threat intelligence feeds

Logging & Auditing

- □ Log all authentication attempts (success/failure)
- □ Audit all financial transactions
- □ Log administrative actions
- □ Monitor file access and modifications
- □ Track API usage patterns
- □ Secure log storage and retention

Incident Response

- □ Develop incident response playbooks
- □ Implement automated threat response
- □ Create security breach notification procedures
- □ Establish forensic data collection processes
- □ Train team on security incident handling
- □ Regular incident response drills

🌐 Infrastructure Security

Server Security

- □ Keep operating systems and software updated
- □ Implement host-based firewalls
- □ Use intrusion prevention systems (IPS)
- □ Regular security patches and updates
- □ Secure server hardening configurations
- □ Monitor system resource usage

Network Security

- □ Implement network segmentation
- □ Use VPNs for administrative access
- □ Deploy DDoS protection
- □ Monitor network traffic for anomalies
- □ Implement zero-trust network architecture
- □ Regular network penetration testing

Cloud Security (if applicable)

- □ Enable cloud security center monitoring
- □ Implement cloud access security broker (CASB)
- □ Secure cloud storage configurations
- □ Monitor cloud resource access
- □ Implement cloud workload protection
- □ Regular cloud security assessments

🔚 Security Testing & Compliance

Regular Security Testing

- □ Conduct monthly vulnerability scans
- □ Perform quarterly penetration testing
- □ Implement automated security testing in CI/CD
- □ Code security reviews for all releases
- □ Third-party security audits annually
- □ Bug bounty program implementation

Compliance & Standards

- □ GDPR compliance for EU users
- □ POPIA compliance for South African users
- □ SOC 2 Type II certification
- □ ISO 27001 compliance assessment
- □ Regular compliance audits
- □ Data retention policy compliance

📋 Security Governance

Policies & Procedures

- □ Develop comprehensive security policies
- □ Create data handling procedures
- □ Implement access control policies
- □ Establish change management procedures
- □ Document security incident procedures
- □ Regular policy reviews and updates

Training & Awareness

- □ Security awareness training for all staff
- □ Phishing simulation exercises
- □ Secure coding training for developers
- □ Regular security briefings
- □ Social engineering awareness
- □ Third-party security requirements

🚀 Deployment Security

Secure DevOps

- □ Implement secrets management in CI/CD
- □ Security scanning in build pipelines
- □ Secure container image scanning
- □ Infrastructure as Code security validation
- □ Environment-specific security configurations
- □ Automated security testing deployment

Production Security

□ Blue-green deployment security validation

□ Production environment hardening

□ Secure configuration management

□ Regular security configuration audits

□ Disaster recovery security considerations

□ Backup security and encryption

📊 Security Metrics & KPIs

Track these security metrics:

- **Authentication**: Failed login attempts, MFA adoption rate
- **Vulnerabilities**: Time to patch, vulnerability severity distribution
- **Incidents**: Mean time to detection (MTTD), mean time to response (MTTR)
- **Compliance**: Audit findings, compliance score
- **Training**: Security training completion rate, phishing test results

🎯 Priority Implementation Order

Phase 1 (Critical - Implement First)

1. Strong password policies and MFA
2. HTTPS/TLS encryption everywhere
3. Input validation and SQL injection prevention
4. Rate limiting and DDoS protection
5. Basic logging and monitoring

Phase 2 (High Priority)

1. Advanced authentication (JWT security, token management)
2. Database and file encryption
3. API security hardening
4. Security monitoring and alerting
5. Mobile app security measures

Phase 3 (Medium Priority)

1. Advanced threat detection
2. Compliance implementations
3. Security testing automation
4. Incident response procedures
5. Security governance frameworks

Remember: Security is not a one-time implementation but an ongoing process that requires regular updates, monitoring, and improvement.