Trading Journal Application – QA & Customer Support Portfolio

This repository showcases my **Quality Assurance (QA)** and **Customer Support** work for **a** trading journal application. It demonstrates how I test features, report bugs, support users, and collaborate with engineering teams in a real SaaS environment.

All examples are **sanitized and representative** of real production workflows. No confidential information, real user data, or proprietary implementation details are included.

About the Application

It is a web and mobile trading journal application used by active traders to track trades, analyze performance, and improve decision-making.

Key capabilities:

- Manual trade logging (stocks, crypto, forex)
- Performance analytics (PnL, win rate, risk metrics)
- Trade notes, tags, and screenshots
- Cloud sync across devices
- Data export (CSV / PDF / HTML)

My Role & Responsibilities

In this project, I worked across **QA and Customer Support**, covering the full product lifecycle:

Quality Assurance

- Manual functional testing (web & mobile)
- Smoke and regression testing before releases
- Test case creation for user-critical flows
- Bug reporting with clear reproduction steps
- Bug verification after fixes

Customer Support

- Ticket triage and categorization
- Issue reproduction and root cause analysis

- Coordination with engineering for defects
- User communication and resolution follow-up
- Knowledge base documentation

What This Repository Contains

This repository is structured to mirror how documentation is handled in real product teams:

- **Test Plan** – Scope, strategy, environments, and exit criteria
- **Test Cases** – Core user flows such as trade creation and editing
- **Bug Reports** – Well-documented, reproducible defects
- **Regression Checklist** – Release readiness validation
- **Support Workflow** – Ticket handling and escalation process
- **Root Cause Analysis** – Post-issue investigation and prevention
- **Knowledge Base Articles** – User-facing help content

Each file is labeled and organized by purpose to make review quick and intuitive.

How to Review This Portfolio

If you are a **recruiter, client, or hiring manager**, here is a suggested review path:

1. Start with this **README** to understand context and scope
2. Review the **Test Plan** to see overall QA strategy
3. Check **Test Cases** for coverage of critical user journeys
4. Read **Bug Reports** to evaluate communication clarity
5. Look at **Support Workflow & RCA** for problem-solving depth

---

Tools & Practices Demonstrated

- Manual QA testing
- Regression testing
- Bug lifecycle management
- Support ticket triage

- Root cause analysis
- Documentation best practices

This portfolio emphasizes **clarity, structure, and user impact**, rather than tooling or automation volume.

Notes on Data & Privacy

- All examples are sanitized
- No real credentials, user data, or internal systems are exposed
- Scenarios reflect common issues in trading journal applications

---

Contact

If you'd like to discuss my QA or Customer Support experience, feel free to reach out via my Upwork or GitHub profile.