## 📑 Mục Lục

1. [Giới thiệu tổng quan](#X1a91e4f164449237285e1b57e3aecdcb689da8d)
2. [Mục tiêu dự án](#Xb0fa18b47f3e5164e4f40bc5add06fa51ae6ce1)
3. [Phân tích hệ thống](#Xcdaa9dd00e3682c78c8efc8fea8834ceb9262f2)
4. [Các chức năng chính](#Xa4a557e3981ce7bdcf84ec600294fdf27a0cbf8)
5. [Phân tích kỹ thuật](#X4ba053f9b222558532235e9f2dff1757d54e12c)
6. [Tích hợp thanh toán VNPay](#X0653207a5274d3021531ba651424ca9ea73d41c)
7. [Triển khai hệ thống](#Xce9a6a6bb8f17d777c0da58f612f390d318ce77)
8. [Kiểm thử & Đánh giá](#X0db511260c01aa34053f0e691d11405df8318fd)
9. [Kết luận & Hướng phát triển](#X316eb1ba8e90276bf7bc2dc7d782572f8abedf4)
10. [Phụ lục](#Xd02330499971ea49625c0bc5860762cb13bbea0)

## 1. Giới thiệu tổng quan

**MantleUR** là nền tảng livestream tương tác hiện đại, tích hợp chat real-time, thanh toán VNPay và hệ thống quản trị người dùng. Dự án hướng tới việc tạo ra trải nghiệm mượt mà, thân thiện và bảo mật cho cả streamer lẫn người xem.

* **Tên dự án:** MantleUR
* **Thể loại:** Ứng dụng Web (Livestream Platform)
* **Sinh viên:** Trịnh Xuân Kiên
* **Website demo:** <https://mantleur-frontend.vercel.app>
* **Backend API:** <https://mantleur.onrender.com>

## 2. Mục tiêu dự án

* Xây dựng nền tảng livestream fullstack (React + Node.js).
* Hỗ trợ thanh toán VNPay thực tế (sandbox).
* Tạo giao diện thân thiện, dễ dùng, responsive.
* Có hệ thống admin quản lý người dùng và stream.
* Đảm bảo bảo mật bằng JWT, mã hóa mật khẩu và phân quyền rõ ràng.

## 3. Phân tích hệ thống

Phần **phân tích hệ thống** (bao gồm các sơ đồ Use Case, Sequence, ERD, Class Diagram) được trình bày chi tiết tại liên kết sau:  
👉 [**Xem chi tiết sơ đồ tại đây**](https://drive.google.com/drive/folders/1rgmJvDBVZONyVTL1TuJDhmqRVAbwjrDT?usp=sharing)

Tóm tắt mô hình hoạt động:

- Người dùng đăng ký → đăng nhập → tham gia stream.  
- Streamer tạo phòng livestream, nhận donate.  
- Người xem tương tác qua chat và gửi tip.  
- VNPay xử lý thanh toán, hệ thống xác thực callback và cập nhật số dư.  
- Admin theo dõi toàn bộ hoạt động và phê duyệt rút tiền.

## 4. Các chức năng chính

### 👤 Người dùng

* Đăng ký, đăng nhập, cập nhật hồ sơ.
* Xem danh sách stream và tham gia xem.
* Chat trực tiếp trong luồng stream.
* Gửi tip/donation thông qua VNPay.

### 🎥 Streamer

* Tạo, quản lý và kết thúc stream.
* Theo dõi lượng người xem, tổng tip.
* Gửi yêu cầu rút tiền.

### 🛠️ Quản trị viên (Admin)

* Quản lý người dùng (ban/kích hoạt).
* Quản lý luồng stream, logs.
* Duyệt giao dịch, rút tiền, thống kê hệ thống.

## 5. Phân tích kỹ thuật

### 5.1. Frontend

* **React 18** – UI framework hiện đại.
* **Vite** – Build tool hiệu năng cao.
* **Tailwind CSS** – CSS framework tiện dụng.
* **Socket.io Client** – Chat real-time.
* **Axios** – Kết nối API.

### 5.2. Backend

* **Node.js + Express.js** – Server chính.
* **Prisma ORM** – Làm việc với database.
* **Socket.io Server** – Quản lý real-time event.
* **JWT + bcryptjs** – Bảo mật & xác thực.

### 5.3. Cơ sở dữ liệu

* **PostgreSQL (Supabase)** – Lưu trữ dữ liệu người dùng, stream, giao dịch.

## 6. Tích hợp thanh toán VNPay

### Quy trình thanh toán

1. Người xem nhập số tiền tip → nhấn “Thanh toán”.
2. Backend tạo URL VNPay với thông tin được mã hóa.
3. VNPay xử lý giao dịch → callback về /api/payments/return.
4. Hệ thống xác thực hash → cập nhật trạng thái thanh toán → cộng tiền.

### Bảo mật

* Dùng VNP\_HASH\_SECRET để tạo mã ký hash xác thực.
* Chỉ chấp nhận callback từ IP hợp lệ của VNPay.
* Mọi giao dịch đều được ghi log để truy vết.

## 7. Triển khai hệ thống

### Hosting

* **Frontend:** Vercel
* **Backend:** Render
* **Database:** Supabase PostgreSQL

### Quy trình triển khai

1. Push code lên GitHub.
2. Vercel/Render tự động build và deploy.
3. Prisma migrate database và khởi tạo dữ liệu mẫu.
4. Kiểm tra API, socket và frontend.

## 8. Kiểm thử & Đánh giá

### Kịch bản kiểm thử

* Đăng ký, đăng nhập ✅
* Tạo và xem stream ✅
* Chat real-time ✅
* Thanh toán VNPay ✅
* Dashboard admin ✅

### Hiệu suất

* **Frontend Load Time:** < 2s
* **Database Response:** < 100ms
* **Uptime Backend:** 99.9%

## 9. Kết luận & Hướng phát triển

### Kết luận

Dự án **MantleUR** đã hoàn thiện nền tảng livestream có tích hợp thanh toán thực tế, bảo mật, và dễ mở rộng. Hệ thống hoạt động ổn định và có khả năng phát triển thành sản phẩm thương mại.

Trong quá trình thực hiện đồ án, tôi đã nắm vững và áp dụng nhiều công nghệ hiện đại như **Node.js & Express** (xây dựng backend, API), **Prisma ORM** (thiết kế schema, quản lý dữ liệu), **React & Vite** (phát triển frontend nhanh), **Supabase** (cơ sở dữ liệu và xác thực), **VNPay Sandbox** (tích hợp thanh toán thử nghiệm), cùng **CI/CD**, **Git & GitHub** giúp tự động hóa triển khai và kiểm soát phiên bản hiệu quả.

Dùng Node.js script (chokidar + simple-git) để auto commit – push – deploy, giúp tiết kiệm thời gian, tránh sót file, đồng bộ liên tục, và hỗ trợ teamwork hiệu quả. Áp dụng CI/CD theo hướng **nhẹ, tự động hóa tại môi trường local** và **đồng bộ trực tiếp với GitHub & Render** để hỗ trợ triển khai backend/frontend nhanh chóng.

Bên cạnh đó, tôi cũng rèn luyện được nhiều **kỹ năng mềm** quan trọng như **làm việc nhóm**, **quản lý thời gian**, **tư duy giải quyết vấn đề**, **kỹ năng viết báo cáo và trình bày**, cùng khả năng **tự học – thích nghi nhanh** với công nghệ mới trong suốt quá trình phát triển hệ thống.

### Hướng phát triển

* Thêm cổng thanh toán (Momo, Stripe,…).
* Hỗ trợ lưu trữ & phát lại video (VOD).
* Gợi ý streamer nổi bật bằng AI.
* Phát triển ứng dụng mobile native.
* Tích hợp thanh toán bằng Stablecoin Crypto

## 10. Phụ lục

### Tài khoản thử nghiệm

Admin: admin@example.com / admin123  
User: testuser@gmail.com / 123456

### Liên hệ

* **Nhóm phát triển:** MantleUR Team
* **GitHub:** <https://github.com/xuankientr/mantleur>
* **Frontend:** <https://mantleur-frontend.vercel.app>
* **Backend API:** <https://mantleur.onrender.com>

*Made with ❤️ by MantleUR Team*