

HƯỚNG DẪN CÀI ĐẶT VÀ CHẠY HỆ THỐNG READING PRACTICE SYSTEM

1. YÊU CẦU HỆ THỐNG

#Software Requirements:

- **Node.js**: Version 16.0 hoặc cao hơn
- **MongoDB** Version 5.0 hoặc cao hơn
- **NPM**: Version 8.0 hoặc cao hơn (đi kèm với Node.js)
- **Git**: Để clone repository

📁 CLONE VÀ CẤU HÌNH DỰ ÁN

1. Clone repository:

Nếu có Git repository

```
git clone https://github.com/TramMai01/tn-DA21TTA-110121062-TramNgocMai-  
HTLTENGLISHREADING.git
```

```
cd reading-practice-system
```

Hoặc nếu có file zip

Giải nén file zip vào thư mục mong muốn

```
cd path/to/extracted/folder
```

2. Cài đặt dependencies:

Cài đặt tất cả packages cần thiết

```
npm install
```

Nếu gặp lỗi, thử:

```
npm install --force
```

hoặc

```
npm ci
```

3. Kiểm tra cấu trúc thư mục:

```
reading-practice-system/
```

```
|— app.js           # Main application file
```

```
|— package.json      # Dependencies và scripts
```

```
|— package-lock.json # Lock file
```

```
|— bin/
```

```
|   └── www.js      # Server startup script
|── config/         # Configuration files
|── controllers/    # Business logic
|── models/         # Database models
|── routes/         # Route definitions
|── views/          # EJS templates
|── public/         # Static assets
|── middleware/     # Custom middleware
└── scripts/       # Utility scripts
```

🌀 CẤU HÌNH ENVIRONMENT VARIABLES

1. Tạo file .env:

Tạo file .env trong thư mục gốc của dự án

touch .env # Linux/macOS

hoặc tạo file .env bằng text editor trên Windows

2. Nội dung file .env:

Database Configuration

MONGODB_URI=mongodb://localhost:27017/testreadingchuan

Session Configuration

SESSION_SECRET=your-super-secret-session-key-here-make-it-long-and-random

Application Configuration

NODE_ENV=development

PORT=3000

Optional: Admin Setup

ADMIN_EMAIL=admin@example.com

ADMIN_PASSWORD=admin123

ADMIN_USERNAME=admin

ADMIN_FULLNAME=System Administrator

3. Tạo Session Secret mạnh:

Tạo random string cho SESSION_SECRET

```
node -e "console.log(require('crypto').randomBytes(64).toString('hex'))"
```

```
# Copy kết quả và thay thế vào SESSION_SECRET trong file .env
```

KHỞI CHẠY HỆ THỐNG

1. Build CSS với Tailwind:

```
# Build CSS files
```

```
npm run tailwind:build
```

```
# Hoặc chạy trong watch mode (tự động rebuild khi có thay đổi)
```

```
npm run tailwind:build -- --watch
```

2. Start MongoDB (nếu chưa chạy):

```
# Windows (nếu không cài as service)
```

```
mongod
```

```
# macOS với Homebrew
```

```
brew services start mongodb/brew/mongodb-community
```

3. Khởi chạy ứng dụng:

```
# Development mode (với nodemon - auto restart)
```

```
npm run dev
```

```
# Production mode
```

```
npm start
```

```
# Hoặc chạy trực tiếp
```

```
node ./bin/www
```

4. Kiểm tra hệ thống:

- Mở trình duyệt và truy cập: <http://localhost:3000>

- Bạn sẽ thấy trang chủ của Reading Practice System

- Nếu thành công, bạn sẽ thấy giao diện chính của hệ thống

TẠO TÀI KHOẢN ADMIN ĐẦU TIÊN

Cách 1: Đăng ký qua giao diện web

1. Truy cập <http://localhost:3000/auth/register>

2. Đăng ký tài khoản mới

3. Sau khi đăng ký, truy cập MongoDB để thay đổi role:

```
# Mở MongoDB shell
mongosh
# Chuyển sang database của ứng dụng
use testreadingchuan
# Tìm user vừa tạo và thay đổi role thành admin
db.users.updateOne(
  { email: "your-email@example.com" },
  { $set: { role: "admin" } }
)
# Kiểm tra kết quả
db.users.findOne({ email: "your-email@example.com" })
```

Cách 2: Tạo script tự động (Khuyến nghị)

Tạo file `scripts/create-admin.js`:

```
const mongoose = require('mongoose');
const bcrypt = require('bcrypt');
const User = require('../models/User');
require('dotenv').config();

async function createAdmin() {
  try {
    await mongoose.connect(process.env.MONGODB_URI);

    const adminData = {
      username: 'admin',
      email: 'admin@system.com',
      password: 'admin123', // Sẽ được hash tự động
      fullName: 'System Administrator',
      role: 'admin',
      isVerified: true
    };
  }
}
```

```

};

const existingAdmin = await User.findOne({ email: adminData.email });
if (existingAdmin) {
  console.log('Admin user already exists!');
  return;
}

const admin = new User(adminData);
await admin.save();

console.log('Admin user created successfully!');
console.log('Email:', adminData.email);
console.log('Password:', 'admin123');
console.log('Please change the password after first login!');

} catch (error) {
  console.error('Error creating admin:', error);
} finally {
  await mongoose.disconnect();
}
}

createAdmin();

```

Chạy script:

```
node scripts/create-admin.js
```

TROUBLESHOOTING

Lỗi thường gặp và cách khắc phục:

1. MongoDB Connection Error

Error: connect ECONNREFUSED 127.0.0.1:27017

****Giải pháp:****

- Kiểm tra MongoDB service đang chạy
- Kiểm tra MONGODB_URI trong file .env
- Restart MongoDB service

Windows

net start MongoDB

macOS

brew services restart mongodb/brew/mongodb-community

2. Port đã được sử dụng

Error: listen EADDRINUSE: address already in use :::3000

****Giải pháp:****

- Thay đổi PORT trong file .env

3. NPM Install Errors

npm ERR! code ERESOLVE

****Giải pháp:****

Xóa node_modules và package-lock.json

rm -rf node_modules package-lock.json # macOS/Linux

rmdir /s node_modules & del package-lock.json # Windows

Cài lại

npm install --legacy-peer-deps

4. CSS không load

****Giải pháp:****

Build lại CSS

npm run tailwind:build

Kiểm tra file public/css/styles.css đã được tạo

5. Session không hoạt động

****Giải pháp:****

- Kiểm tra SESSION_SECRET trong .env
- Clear browser cache và cookies
- Restart server