QUICK SETUP GUIDE

NUMBER-1: FULL-STACK WEB APP (EXPRESS + VUE)

Quick Start:

• Navigate to the project directory:

cd number-1

• Install all dependencies:

npm run install:all

• Start the development server:

npm run dev

Access URLs:

• Frontend: http://localhost:5173

• Backend: http://localhost:3000

API Endpoints:

• POST /api/submit-data - Submit form data

• GET /api/get-data - Get all stored data

• PATCH /api/update-data/:id - Update specific entry

• DELETE /api/delete-data/:id - Delete specific entry

NUMBER-2: DATA AUTOMATION SCRIPTS (PYTHON + CRON)

Quick Start:

• Navigate to scripts directory:

cd number-2/home/scripts

• Make scripts executable:

chmod +x \*.py \*.sh

• Run setup script:

./setup\_cron.sh

Cron Schedule:

• 08:00, 12:00, 15:00 WIB - Data collection

• 02:00 WIB - Data cleanup (30-day retention)

Manual Testing:

• Test data collection:

python3 data\_collection/collect\_data.py

• Test data cleansing:

python3 data\_cleansing/clean\_data.py

NUMBER-3: SQL DATABASE OPERATIONS

Quick Start:

• Navigate to SQL directory:

cd number-3

• Run scripts in order:

mysql -u user -p db < 01\_insert\_employee.sql

mysql -u user -p db < 02\_update\_engineer\_salaries.sql

mysql -u user -p db < 03\_calculate\_2021\_expenditure.sql

mysql -u user -p db < 04\_top\_experienced\_employees.sql

mysql -u user -p db < 05\_engineer\_experience\_filter.sql

Scripts Overview:

1. 01\_insert\_employee.sql - Add Albert (engineer, $50 salary)

2. 02\_update\_engineer\_salaries.sql - Set all engineers to $85 salary

3. 03\_calculate\_2021\_expenditure.sql - Calculate 2021 total costs

4. 04\_top\_experienced\_employees.sql - Get top 3 most experienced employees

5. 05\_engineer\_experience\_filter.sql - Find engineers with ≤3 years experience

SYSTEM REQUIREMENTS

Number-1 Requirements:

• Node.js version 16 or higher

• npm or yarn package manager

Number-2 Requirements:

• Python 3.6 or higher

• Linux/Unix system with cron support

• Bash shell

Number-3 Requirements:

• Any SQL-compliant database (MySQL, PostgreSQL, SQLite)

• Database client or command-line tool

TROUBLESHOOTING

Number-1 Issues:

• Port already in use:

netstat -ano | findstr :3000

• Clear npm cache:

npm cache clean --force

• Reinstall dependencies:

rm -rf node\_modules package-lock.json

npm install

Number-2 Issues:

• Check cron jobs:

crontab -l

• Check collection logs:

tail -f /home/cron/logs/collection.log

• Check cleansing logs:

tail -f /home/cron/logs/cleansing.log

• Verify script permissions:

ls -la /home/scripts/data\_collection/collect\_data.py

Number-3 Issues:

• Test database connection:

mysql -u user -p -e "SELECT 1;"

• Check if table exists:

DESCRIBE employee;

• Verify database exists:

SHOW DATABASES;

PROJECT STRUCTURE

Test-Huawai/

├── number-1/ (Full-stack web application)

│ ├── client/ (Vue.js frontend)

│ │ ├── src/

│ │ │ ├── App.vue (Main application component)

│ │ │ ├── main.js (Vue app entry point)

│ │ │ └── style.css (Global styles)

│ │ ├── public/

│ │ ├── package.json (Frontend dependencies)

│ │ └── vite.config.js (Vite configuration)

│ ├── server/

│ │ └── index.js (Express.js server)

│ ├── package.json (Backend dependencies)

│ ├── config.env (Environment variables)

│ └── README.md

├── number-2/ (Python automation scripts)

│ └── home/scripts/

│ ├── data\_collection/

│ │ └── collect\_data.py (Data collection script)

│ ├── data\_cleansing/

│ │ └── clean\_data.py (Data cleansing script)

│ ├── setup\_cron.sh (Cron setup script)

│ ├── test.py (Test script)

│ └── README.txt

└── number-3/ (SQL database operations)

├── 01\_insert\_employee.sql

├── 02\_update\_engineer\_salaries.sql

├── 03\_calculate\_2021\_expenditure.sql

├── 04\_top\_experienced\_employees.sql

├── 05\_engineer\_experience\_filter.sql

└── README.md

NOTES:

• All commands should be run from the respective project directories

• Ensure proper permissions are set for script execution

• Database credentials should be configured before running SQL scripts

• Log files are automatically created in /home/cron/logs/ for automation scripts

• The web application uses in-memory storage (data resets on server restart)