

Hướng Dẫn Phát Triển và Triển Khai

Mục Lục

- 1. Yêu Cầu Hệ Thống
- 2. Cài Đặt và Thiết Lập
- 3. Quy Trình Phát Triển
- 4. Kiểm Thử
- 5. Triển Khai
- 6. Quy Tắc và Tiêu Chuẩn

Yêu Cầu Hệ Thống

- Node.js (phiên bản mới nhất)
- Bun Runtime
- Git
- VS Code (khuyến nghị)

VS Code Extensions Cần Thiết

- Tailwind CSS IntelliSense
- ES7+ React/Redux/React-Native snippets
- Biome Extension

Cài Đặt và Thiết Lập

1. Clone dự án:

```
git clone [repository-url]
cd react-vite-with-unit-test
```

2. Cài đặt dependencies:

```
bun install
```

3. Thiết lập môi trường:

- Sao chép file `.env.example` thành `.env.development`
- Cập nhật các biến môi trường cần thiết

4. Thiết lập Git hooks:

```
bun prepare
```

Quy Trình Phát Triển

1. Khởi Động Môi Trường Phát Triển

```
bun dev
```

2. Phát Triển Component

1. Tạo component mới:

```
bun generate
```

2. Phát triển trong Storybook:

```
bun storybook
```

3. Quy Trình Code

1. Tạo nhánh mới:

```
git checkout -b feature/ten-tinh-nang
```

2. Format code trước khi commit:

```
bun format
bun lint
```

3. Kiểm tra code:

```
bun check
```

Kiểm Thử

1. Unit Tests

```
# Chạy tất cả tests
bun test

# Chạy tests với watch mode
bun test:watch

# Kiểm tra coverage
bun test:coverage
```

2. Storybook Tests

```
# Chạy Storybook
bun storybook

# Build Storybook
bun build-storybook
```

Triển Khai

1. Build Production

```
bun build
```

2. Preview Build

```
bun preview
```

3. Docker Deployment

```
# Build Docker image
docker build -t react-vite-app .

# Run container
docker run -p 8080:80 react-vite-app
```

Quy Tắc và Tiêu Chuẩn

1. Code Style

- Sử dụng Biome cho formatting và linting
- Tuân thủ TypeScript strict mode
- Sử dụng functional components và hooks

2. Git Commit

- Sử dụng conventional commits
- Mỗi commit phải pass tất cả tests
- Pre-commit hooks sẽ kiểm tra:
 - Code formatting
 - Linting
 - Type checking
 - Unit tests

3. Component Development

1. Tổ chức thư mục:

```
src/
├── components/
│   ├── [ComponentName]/
│   │   ├── index.tsx
│   │   ├── [ComponentName].stories.tsx
│   │   └── [ComponentName].test.tsx
│   └── ...
├── hooks/
├── utils/
└── ...
```

2. Component Guidelines:

- Sử dụng TypeScript interfaces cho props
- Implement error boundaries
- Tối ưu performance với React.memo khi cần thiết
- Viết stories cho tất cả các cases

4. Testing Guidelines

- Coverage tối thiểu: 80%
- Test các edge cases
- Mock external dependencies
- Sử dụng testing-library best practices

5. Performance Optimization

- Lazy loading cho routes
- Code splitting
- Image optimization
- PWA implementation

CI/CD Pipeline

GitHub Actions workflow bao gồm:

1. Build check

2. Unit tests
3. Storybook build
4. Docker image build
5. Deployment to staging/production

Monitoring và Logging

1. Error tracking với Error Boundary
2. Performance monitoring
3. User analytics

Security Guidelines

1. Dependency scanning
2. Regular updates
3. Security best practices
4. Environment variables management

Support

Nếu có vấn đề:

1. Kiểm tra documentation
2. Tạo issue trên GitHub
3. Liên hệ team lead