

1. 서론: AI 기반 UI/UX 우수성 및 Dash Mantine Components 가이드

- **목표:** Dash Mantine Components (DMC)를 활용하여 깔끔하고 유용한 Dash 애플리케이션 개발을 위한 가이드 제공
- **핵심:** "The UI_UX Playbook"의 UI/UX 모범 사례를 AI 기반 Dash 개발 지침으로 전환
- **UI/UX 상호작용:** 고품질 UI/UX는 디자인 원칙의 세심한 적용 결과이며, UI는 UX의 분위기를 설정, UI와 UX 모두 능숙해야 최적의 사용자 경험 달성
- **AI를 위한 선언적 안내:** 좋은 UI의 설명과 더불어 특정 DMC 코드 패턴과 속성 구성을 상세히 설명하여 AI에게 명확한 지침 제공
- **범위:** dash==3.0.4 버전과 해당 DMC 버전에 중점, 향후 업데이트 시 수정 필요
- **주요 영역:** 테마, 레이아웃, 데이터 표시, 타이포그래피, 데이터 시각화

2. Dash Mantine Components를 이용한 핵심 테마

• dmc.MantineProvider를 이용한 테마 관리

- **dmc.MantineProvider:** 모든 DMC 애플리케이션을 감싸는 핵심 컴포넌트
- **역할:** 테마 전파, 색 구성표 관리, 전역 스타일 및 CSS 변수 주입
- **주요 속성:** theme (테마 객체), forceColorScheme (색 구성표 설정), withGlobalStyles (Mantine 기본 스타일 적용), withNormalizeCSS (CSS 정규화), inherit (부모 요소 스타일 상속)

• 색상 시스템 정의

- **색상 사용 원칙:** 절제, 배경 선택, 대비, 의미론적 색상 사용
- **DMC 구현:** theme.primaryColor (주요 강조 색상), theme.colors (사용자 정의 색상 팔레트), theme.primaryShade (primaryColor의 음영 지정)

- **AI 지침:** 애플리케이션 브랜드 아이덴티티 기반 primaryColor 정의, theme.colors에 10개의 음영을 가진 브랜드 색상 정의, primaryShade 조정을 통해 밝은/어두운 모드에서 대비 유지, color 속성을 통한 컴포넌트 색상 설정, Mantine CSS 변수 활용, theme.autoContrast 및 theme.luminanceThreshold 사용

• 타이포그래피 시스템

- **타이포그래피 원칙:** 글꼴 선택, 줄 길이, 텍스트 대비, 글꼴 두께, 크기 및 계층 구조, 줄 높이, 텍스트 정렬
- **DMC 구현:** theme.fontFamily (전역 글꼴 모음), theme.fontFamilyMonospace (모노스페이스 글꼴), theme.headings (제목 스타일링), theme.fontSizes (미리 정의된 글꼴 크기), theme.lineHeights (미리 정의된 줄 높이)
- **AI 지침:** 가독성 높은 산세리프 글꼴 사용, theme.headings.sizes 구성을 통한 시각적 계층 구조 설정, theme.fontSizes.md를 16px로 설정, theme.lineHeights.md를 1.5~1.6으로 설정

• 간격, 곡선 및 그림자

- **원칙:** 여백, 깊이 및 질감, 모양 일관성
- **DMC 구현:** theme.spacing (간격 토큰), theme.radius (테두리 반경 토큰), theme.defaultRadius (기본 테두리 반경), theme.shadows (박스-그림자 토큰)
- **AI 지침:** theme.spacing을 기본 단위의 배수로 정의, defaultRadius 설정, theme.shadows에 미묘한 그림자 시스템 정의

• 전역 컴포넌트 스타일링

- **원칙:** 일관성, 단순성
- **DMC 구현:** theme.components (전역 컴포넌트 사용자 정의), defaultProps 설정, 전역 styles 또는 classNames 적용 (Styles API)

- **AI 지침:** 자주 사용되는 컴포넌트에 대한 defaultProps 정의, theme.components.ComponentName.styles를 사용한 전역 스타일링

3. Dash Mantine Components의 레이아웃 및 구조

• 시각적 계층 구조 달성

- **DMC 구현:** dmc.Title (order, size, fw 속성), dmc.Text (size, fw, c 속성), dmc.Button (variant, color 속성), dmc.Card (shadow 속성)
- **AI 지침:** dmc.Title을 사용한 제목 설정, dmc.Text를 사용한 본문 텍스트 작성, dmc.Button의 variant 및 color를 통한 버튼 계층 구조 설정, dmc.Card의 shadow를 통한 중요도 표현, 색상 및 대비 유지, 크기 및 위치 고려

• 그리드 시스템 및 반응형 레이아웃

- **DMC 구현:** dmc.Grid, dmc.GridCol (columns, gutter, justify, align, grow, span, offset, order 속성), dmc.SimpleGrid (cols, spacing, verticalSpacing 속성)
- **AI 지침:** dmc.Grid를 사용한 주요 콘텐츠 영역 구조화, theme.spacing 사용, dmc.GridCol의 반응형 span 속성 사용, dmc.SimpleGrid를 사용한 동일 크기 항목 정렬

• 요소 쌓기 및 그룹화

- **DMC 구현:** dmc.Stack (gap, align, justify 속성), dmc.Group (gap, align, justify, grow, wrap 속성), dmc.Flex (direction, wrap, align, justify, gap, rowGap, columnGap 속성)
- **AI 지침:** dmc.Stack을 사용한 요소의 세로 배열, dmc.Group을 사용한 요소의 가로 배열, dmc.Flex를 사용한 복잡한 레이아웃 구현, theme.spacing 사용

• 간격 및 포함

- **DMC 구현:** dmc.Space (h, w 속성), dmc.Container (size, fluid 속성), dmc.Center (inline 속성), dmc.AspectRatio (ratio 속성)

<ul style="list-style-type: none">◦ AI 지침: dmc.Space를 사용한 여백 추가, dmc.Container를 사용한 콘텐츠 최대 너비 제어, dmc.Center를 사용한 요소 중앙 배치, dmc.AspectRatio를 사용한 종횡비 유지
<ul style="list-style-type: none">• 애플리케이션 셀<ul style="list-style-type: none">◦ DMC 구현: dmc.AppShell (padding, layout, zIndex, withBorder, header, navbar, aside, footer 속성)◦ AI 지침: dmc.AppShell을 사용한 애플리케이션 구조 설정, 반응형 동작을 위한 속성 구성
4. 데이터 표시 및 피드백
<ul style="list-style-type: none">• 이미지 표시<ul style="list-style-type: none">◦ DMC 구현: dmc.Image (src, alt, fit, radius, h, w, withPlaceholder, placeholder, caption 속성)◦ AI 지침: 설명적인 alt 속성 제공, fit 속성 사용, h 및 w 속성 사용, withPlaceholder 사용, caption 사용
<ul style="list-style-type: none">• 목록 및 테이블<ul style="list-style-type: none">◦ DMC 구현: dmc.List, dmc.ListItem, dmc.Table◦ AI 지침: dmc.List 및 dmc.ListItem을 사용한 목록 생성, dmc.Table을 사용한 테이블 생성, 대용량 데이터의 경우 dash-ag-grid 사용
<ul style="list-style-type: none">• 피드백 및 상태 표시<ul style="list-style-type: none">◦ DMC 구현: dmc.Alert (title, children, color, icon, variant, withCloseButton, duration 속성), dmc.Badge, dmc.Loader, dmc.Progress, dmc.RingProgress, dmc.Themelcon◦ AI 지침: dmc.Alert를 사용한 피드백 제공, 색상 규칙 준수, dmc.Badge, dmc.Loader, dmc.Progress, dmc.RingProgress, dmc.Themelcon 사용
5. 데이터 시각화: Dash Mantine Components의 차트 및 그래프

- **차트 사용 원칙:** 올바른 차트 유형 선택, 명확성 및 단순성, 색상 사용, 상호 작용, 반응성
- **DMC 차트 컴포넌트:** dmc.AreaChart, dmc.BarChart, dmc.LineChart, dmc.PieChart, dmc.DonutChart, dmc.RadarChart, dmc.ScatterChart, dmc.Sparkline, dmc.CompositeChart
- **AI 지침:** 차트 데이터 구조 확인, theme.colors 사용, 타이포그래피 및 접근성 원칙 준수, 설명적인 제목 추가

6. Dash Mantine Components의 타이포그래피

<ul style="list-style-type: none">• 핵심 타이포그래피 원칙<ul style="list-style-type: none">◦ 원칙: 글꼴 선택 및 가독성, 계층 구조, 줄 길이 및 높이, 대비, 정렬
<ul style="list-style-type: none">• DMC 타이포그래피 컴포넌트<ul style="list-style-type: none">◦ dmc.Title: 의미론적 HTML 제목 렌더링 (order, size, fw, c, ta, lineClamp, transform, variant 속성)◦ dmc.Text: 단락 및 일반 텍스트 렌더링 (size, fw, c, ta, span, inherit, gradient, variant, lineClamp, truncate, inline 속성)◦ dmc.Blockquote: 인용된 텍스트 표시 (children, color, icon, cite 속성)◦ dmc.Code: 인라인 또는 블록 수준 코드 스니펫 표시 (children, color, block 속성)◦ dmc.Highlight: 텍스트 내 부분 문자열 강조 (children, highlight, highlightColor, color 속성)◦ AI 지침: dmc.Title을 사용한 제목 설정, dmc.Text를 사용한 본문 텍스트 작성, dmc.Blockquote, dmc.Code, dmc.Highlight 사용

7. 신규 문서화 및 업데이트된 컴포넌트 (v1.3.0)

- **새로운 컴포넌트:** dmc.Table, dmc.TableScrollContainer, dmc.Carousel, dmc.CarouselSlide, dmc.Timeline, dmc.Timelineltem, dmc.Stepper, dmc.StepperStep, dmc.CodeHighlight, dmc.CodeHighlightTabs, dmc.InlineCodeHighlight, dmc.SegmentedControl, dmc.List, dmc.ListItem, dmc.Center, dmc.Menu, dmc.MenuTarget, dmc.MenuDropdown, dmc.MenuItem, dmc.ActionIcon, dmc.Highlight, dmc.Pagination, dmc.Tree, dmc.Grid, dmc.GridCol

8. 이 가이드의 유지 관리 및 모범 사례

- **권장 사항:** 버전 고정, 공식 문서와의 일관성 유지, 커뮤니티 기여 유도, 실용적인 예시 제공, Styles API 문서화

9. 문서 요약 및 특정 섹션 추출

- **전체 문서 요약:** 각 섹션별 내용 요약
- **2. 핵심 테마 요약:** dmc.MantineProvider, 색상 시스템, 타이포그래피 시스템, 간격, 곡선, 그림자, 전역 컴포넌트 스타일링에 대한 설명
- **3. 레이아웃 및 구조 요약:** 시각적 계층 구조, 그리드 시스템, 요소 쌓기 및 그룹화, 간격 및 포함, 애플리케이션 셀에 대한 설명
- **Dash Mantine Components 관련 베스트 프랙티스:** 테마 시스템의 일관된 사용, 레이아웃 및 구조의 체계화, 데이터 표시 및 피드백의 효율성, 데이터 시각화의 명확성, 타이포그래피의 전략적 구현에 대한 모범 사례

10. 마크다운 문서를 기준으로 프로젝트 설계 구조 제안

- **제안하는 프로젝트 구조:** app.py, index.py, config/, assets/, components/, pages/, themes/, utils/, services/, tests/, Dockerfile, requirements.txt, .env, .pylintrc, pyproject.toml, README.md
- **설계 원칙:** 객체 지향 프로그래밍 (OOP) 및 SOLID 원칙 적용, 모듈화 및 재사용성, 유지보수성 및 확장성, TDD (테스트 주도 개발) 지원

11. Python 코드와 연결된 UI 설계 가이드 추출

• **핵심 테마 설정**

- **코드 예시:** main_theme.py (APP_THEME 딕셔너리 정의, primaryColor, colors, fontFamily, headings, fontSizes, lineHeights, spacing, radius, defaultRadius, shadows, components 설정)
- **설명:** 브랜드 색상 정의, 기본 음영 최적화, 전역 글꼴 및 제목 설정, 일관된 간격 및 곡선, 컴포넌트 기본값 설정

• **시각적 계층 구조를 위한 컴포넌트 사용**

- **코드 예시:** dmc.Title, dmc.Text, dmc.Button 사용 예시
- **설명:** 제목 설정, 본문 텍스트 작성, 버튼 계층 구조 설정

• **반응형 레이아웃 구현**

- **코드 예시:** dmc.Grid, dmc.GridCol, dmc.SimpleGrid, dmc.Stack, dmc.Group 사용 예시
- **설명:** 반응형 span 속성 사용, 균등한 크기 항목 정렬, 일관된 간격 유지

• **애플리케이션 셸 구조화**

- **코드 예시:** dmc.AppShell 사용 예시
- **설명:** 헤더, 내비게이션, 푸터 설정, 반응형 동작 구성

• **데이터 시각화 및 피드백**

- **코드 예시:** dmc.Alert 사용 예시
- **설명:** 성공 및 오류 메시지 표시