**Flutter E-Commerce App Documentation**

**Packages Used**

1. **carousel\_slider (^4.2.1)**
   * Purpose: Implements the banner slider on the home page
   * Features: Auto-play, infinite scroll, custom transitions
2. **http (^1.1.0)**
   * Purpose: Handles API requests to fakestoreapi.com
   * Features: Async data fetching, HTTP client functionality

**Design Choices**

**Architecture**

* Structured the app using a feature-based organization
* Separated models, widgets, and services into distinct folders
* Used stateful widgets for components requiring state management

**UI/UX Decisions**

1. **Navigation**
   * Implemented drawer menu for main navigation
   * Used TabBar for product filtering
   * Added persistent header for better usability
2. **Product Display**
   * Grid layout with 2 columns for optimal screen usage
   * Card design with shadow for depth
   * Hero animation for product image transitions
3. **Color Scheme**
   * Primary: #6C5CE7 (Purple)
   * Background: #F8F9FA (Light Gray)
   * Text Colors: #2D3436 (Dark Gray)
   * Used opacity variations for visual hierarchy

**Performance Optimizations**

1. **Image Loading**
   * Implemented loading indicators for network images
   * Used Hero widgets for smooth transitions
   * Optimized image sizes in grid view
2. **Scrolling**
   * Implemented custom scroll behavior
   * Added pull-to-refresh functionality
   * Used SliverGrid for better scroll performance

**Development Challenges**

1. **API Integration**
   * Challenge: Handling network errors and loading states
   * Solution: Implemented comprehensive error handling and loading UI states
2. **Responsive Design**
   * Challenge: Maintaining consistent layout across different screen sizes
   * Solution: Used flexible widgets and responsive measurements
3. **Performance**
   * Challenge: Smooth scrolling with network images
   * Solution: Implemented lazy loading and optimized grid view
4. **State Management**
   * Challenge: Managing product data and UI states
   * Solution: Used StatefulWidget with proper state initialization