Type: Feature	**Title:** Frontpage - Blog	**Status:** ON

Context: Backend features were implemented and now the immediate concerns are parsing the available posts and conveniently adding them to the front page.

Core Decision: Implement a frontpage that has components featuring new, sortable posts

Prospects	Decision Parameters	0TD Concerns
1. Seamless filtration of data during query 2. Limit number of posts on view 3. Three layouts • Card, features description • Bicolumnar grid • Tricolumn grid 4. Functional navbar • By default, posts are sorted by recent • Recency also operates off sort feature 5. Animate on scroll 6. Dark mode saved using cook-ieStore	§ **Animation:** • Use intersection observer API ◦ Slide from below effect (scrolling down) ◦ Slide from above effect (scrolling up) § **Querying:** • Add optional parameter in server action ◦ Introduce optional filtering in query ◦ Encapsulate abstraction in API (RFC) • Use native array map method to display posts § **Navbar:** • Button: **Recent** ◦ Due to sort by recency, this is useless ◦ Transform to: *Choice*, writer's choice for public read · Set specific styles if target blog is in view • Button: **Posts** ◦ Transform to: **Categories** · Shows a succinct list of categories and post count · Use a separate route which nests category routes spread-ing relevant pages • Button: **About** ◦ Uses own route ◦ Ambiguously define goals and intent ◦ Links to relevant platforms	↯ Sort feature not implemented, however with queries API is not difficult ↯ Database query executes n times where $n = total\ post\ count$ ◦ Consider dynamic, on-demand query approach ◦ For seamless ∩ observer API, fetch based on 3x screen length and preload on scroll ◦ Query database when intersecting 79% of the content

Decision Process	Results and Conflicts
1. Removed responsibility of layout grids	