Google Summer of Code

Project Proposal

Build a COSMIC based wayland session for Regolith

COSMIC

Organization	**:**	**CCExtractor**
Mentor	**:**	**Soumya Ranjan Patnaik**
Applicant	**:**	**Sandeep Patel**

CONTACT INFORMATION

NAME : Sandeep Patel
EMAIL : sandeeppatel626735@gmail.com
GITHUB : sandptel
LINKEDIN : sandppatel
LOCATION : Kharagpur, West Bengal, India
TIMEZONE : IST(UTC + 5:30)

STUDENT AFFILIATION

UNIVERSITY : Indian Institute of Technology Kharagpur
DEGREE : Dual Degree Course (B.Tech + M.Tech)
GRADUATION YEAR : 2026

BRIEF BIO

I am Sandeep Patel, a fourth-year undergraduate student at IIT Kharagpur, and have been a **Google Summer of Code 24 Mentee** at CCEXTRACTOR for the project "Create nix derivations for Regolith Desktop". During which I ported regolith-wayland completely to nix and have released a working alpha version of the software by the name regolith-nix. In the same process, I have become a contributor to nixpkgs and a maintainer of 10 packages (mostly regolith packages). Given that I am deeply passionate about technology, particularly Linux/Rust/Nix or and deeply appreciate the fundamentals of Open Source. I have used and even at this point packaged desktop environments but never written applications for desktop environments or had hands-on experience with the compositors ipc or source code. Similar to previous year, this project again represents a significant step beyond my usual activities, pushing me to expand my knowledge and technical skills. Looking forward to contributing again to the regolith project.

Previous Contributions

I made 9 Pull Requests to nixpkgs repository (19.9k⭐) 7 of which are just to directly add **regolith-desktop** related packages that I created during my period of GSOC-24. I received reviews mostly regarding formatting issues and some to enforce nixpkgs idioms on the packages code. Nixpkgs receives a heavy inflow of package merge PRs, delaying the review/merging process. Due to this they often merge reviewed pull requests in batches(my first pull request without any requested change took 3 weeks to merge to master). This is the summary of pull requests I created to add regolith packages to nixpkgs.

Nixpkgs (github: nixos/nixpkgs)

Pull Requests	Status	Review Status	Notes
ilia: init at 3.1	Open	Unreviewed	Added ilia to nixpkgs
trawl: init at 0.2.5	Open	Ready to Merge	Added trawl to nixpkgs
vtracer: init at 0.6.4	Closed	Merged	Added vtracer to nixpkgs
i3xrocks: init at 1.3.6-1	Open	Ready to Merge	Added i3xrocks to nixpkgs
regolith-rofication: init at 1.5.0	Open	Ready to Merge	Added rofication to nixpkgs
nixos/fprintd: add elanmoc2 support	Open	WIP	Updated option to add my personal-laptop's fingerprint sensor's package
i3-swap-focus: init at 0.4.5	Open	Ready to Merge	Added i3-swap-focus to nixpkgs
pulsemeeter: init at 1.2.14	Open	WIP	Added pulsemeter to nixpkgs
xrescat: init at 1.0	Open	Ready to Merge	Added xrescat to nixpkgs
libtrawldb: init at 0.1-3	Open	Ready to Merge	Added libtrawldb to nixpkgs

Project Description

Regolith is a productivity-focused Ubuntu derivative and desktop environment supporting Wayland and Xorg (i3). It integrates lightweight window managers with gnome-session, reducing GNOME Shell complexity while maintaining system management capabilities. Regolith uses GNOME dependencies for GUI features like the control center over tiling window managers (Sway, i3).

However, GNOME's reliance on Mutter and its reliance on custom Wayland protocols (e.g., gtk-shell and xdg-foreign) that aren't implemented in other wlroots-based compositors (sway) limits its modularity. To enhance flexibility, we aim to integrate the newly released cosmic desktop components (cosmic-epoch) which is still an incomplete alpha, yet it offers a more direct and low-threshold approach for seamless integration.

Technical Background

Regolith Desktop

Regolith Desktop is a streamlined, keyboard-centric desktop environment that integrates the i3/sway tiling window manager with many essential components to deliver a cohesive and efficient user experience. In its 3.0 release, Regolith introduced a Wayland-based session utilizing the Sway compositor, expanding compatibility to modern display protocols. This integration allows users to choose between X11 and Wayland sessions, accommodating diverse hardware and preferences. Currently Regolith DE (wayland) utilises on the following components:

- sway-regolith → fork of SwayWM used to with support for Xresources like configuration system
- ilia → a desktop executor/launcher/notification-viewer/file-finder
- regolith-control-center → fork of gnome-control-center adapted to needs of regolith desktop providing display/input/power GUI options
- regolith-session → session scripts+configurations+systemd-services
- regolith-displayd → display-config Dbus daemon for gnome-control-center
- regolith-inputd → input management daemon for integration with gnome-control-center
- regolith-powerd → daemon syncing GSD power settings for sway
- trawl → Xresources like file based configuration system backend
- rofication → notification daemon for regolith to provides a rofi front-end

Cosmic Desktop Environment

System76, known for Pop!_OS, is developing the COSMIC Desktop Environment, fully rewritten in Rust for stability and performance. Unlike its GNOME-based predecessor, the new COSMIC uses the iced-rs toolkit for efficiency and type safety. Recently, System76 released its core components under the cosmic-epoch repository which, although an alpha preview, enables seamless integration with any Wayland-based environment. This modular approach enhances flexibility, making COSMIC a cutting-edge Linux desktop solution allowing anyone to reuse their work as a toolkit rather than a product for further development of Desktop Environments in the Open Source Community. Overview of currently offered cosmic components :

- Cosmic-applets → applets for cosmic-panel
- Cosmic-applibrary → grid based application launcher
- Cosmic-bg → session service which applies backgrounds to displays.
- Cosmic-comp → compositor for cosmic desktop
- Cosmic-edit → text editor
- Cosmic-files → file manager
- Cosmic-greeter → login-manager daemon for cosmic's compositor
- Cosmic-icons → official System76 icon theme
- Cosmic-launcher → frontend for pop-launcher [yet another launcher :)]
- Cosmic-notifications → notifications daemon
- Cosmic-osd → graphic-overlay (volume/brightness sliders)
- Cosmic-panel → configurable status-bar+applets and a dock
- Cosmic-player → a media player
- Cosmic-randr → utility to display and configure wayland outputs?
- Cosmic-screenshot → a wayland screenshot utility
- Cosmic-session → session files (distribution with changes not allowed)
- Cosmic-settings → settings application (display/input/system-states)
- Cosmic-settings-daemon → daemon for cosmic-settings functionality
- Cosmic-store → application store (currently only for debian distros)
- Cosmic-term → terminal emulator based on alacritty
- Cosmic-workspaces-epoch → tool/applet to manager virtual workspaces
- Xdg-desktop-portal-cosmic → cosmic's version of xdg-desktop-portal

Currently What Works & What Doesn’t?

Results from my testing of cosmic-components on regolith-wayland session, a compiled, detailed and most recent version can be found here.

Testing Environment

OS → Ubuntu 24.04.02
Cosmic Installation Method → `sysext` installation

Installation Process

Installation of dependencies on Ubuntu 24.04 via apt :

```
sudo apt install just rustc libglvnd-dev libwayland-dev libseat-dev
libxkbcommon-dev libinput-dev udev dbus libdbus-1-dev libsystemd-dev
libpixman-1-dev libssl-dev libflatpak-dev libpulse-dev pop-launcher
libexpat1-dev libfontconfig-dev libfreetype-dev mold cargo libgbm-dev
libclang-dev libpipewire-0.3-dev libpam0g-dev -y
```

List of some extra dependencies that might be required to install

```
sudo apt install libdisplay-info-dev
sudo apt install gstreamer-sys
sudo apt install libgstreamer1.0-dev gstreamer1.0-plugins-base
gstreamer1.0-plugins-good
sudo apt install libgstreamer-plugins-base1.0-dev
```

Then cloning the repository and compiling all the applications along with `sysext` using these commands

```
git clone --recurse-submodules https://github.com/pop-os/cosmic-epoch
cd cosmic-epoch
just sysext
```

Then copying the `sysext` directory to `/var/lib/extensions` and enabling the `systemd-sysext` and refreshing/rebooting.

```
cp -r cosmic-sysext /var/lib/extensions
sudo systemctl enable --now systemd-sysext
sudo systemd-sysext refresh
```

This currently installs cosmic-desktop as an alternative session in display manager and makes all the cosmic components listed above to the shell.

Components Wise Analysis

This is a simple analysis of what components work and what don't and an expected reason of why it is so? Along with a video-link/attachments demonstrating the breakage.

Cosmic Launcher

Issue

Completely Unresponsive to execution

Reason

Unable to integrate with pop-launcher dependency which is only able to initiate with cosmic-compositor running.

Cosmic Settings

Video Demonstration → 1
Video Demonstration → 2

What works?

1. Display Settings Page
2. Sound Settings Page
3. Desktop Section
   a. Dock Settings (Glitches with auto hide options)
   b. Panel Settings (Glitches with placements)
   c. Appearance with changing accent colors, gaps size & active window hint size settings work (works with cosmic-ext-sway-daemon)

What doesn't?

1. Input Devices Settings
   a. Mouse settings including primary button change, mouse-cursor speed and scrolling speed change are placebo buttons and work only under cosmic-compositor
   b. All Touchpad options buttons and scroll bars are placebo buttons
2. Power Settings (Idle Settings/ Power Mode)
3. Accessibility Settings (Magnifier)
4. Desktop Section
   a. The Window Management Section has all pseudo buttons.
   b. The Workspaces Section does not work as well.
5. Firmware Page in About Section
   a. workspaces section (can use sway ipc rust to fetch workspaces info

⟶ can be implemented right away)

b. window management settings (This sections might need complete redesign to comply with the way currently regolith handles these options)

1. minimize/maximize button visibility and actions
2. Focus change settings

Cosmic Panel

[Video Demonstration]

What works?

1. cosmic-panel integration with cosmic-settings (Changing opacity/ changing placement on screen for dock and panel/ adding and removing applets)

What doesn't?

1. only cosmic-applet-power respects the sway's drawing area. Rest (cosmic-applet-battery/audio/network/input-sources ..etc) violate the rules and span across multiple workspaces.
2. sway workspaces are not detected (expected)
3. cosmic-launcher is non-responsive
4. accessibility settings (magnifier) do not work (non-responsive)

cosmic notifications

Issue

Unresponsive in sway but works perfectly fine in cosmic-comp.

Reason

There is concrete evidence (custom notification implementations under cosmic-protocols rust library) that cosmic-notifications rely on protocols hard coded within the COSMIC compositor (cosmic-comp). The COSMIC ecosystem utilizes custom Wayland protocols, which are implemented specifically in cosmic-comp and not in other compositors like Sway.

gnome-files -> cosmic files

Note: Works Fine with nominal errors

gedit -> cosmic edit

Note: Works Fine

Plan of Action

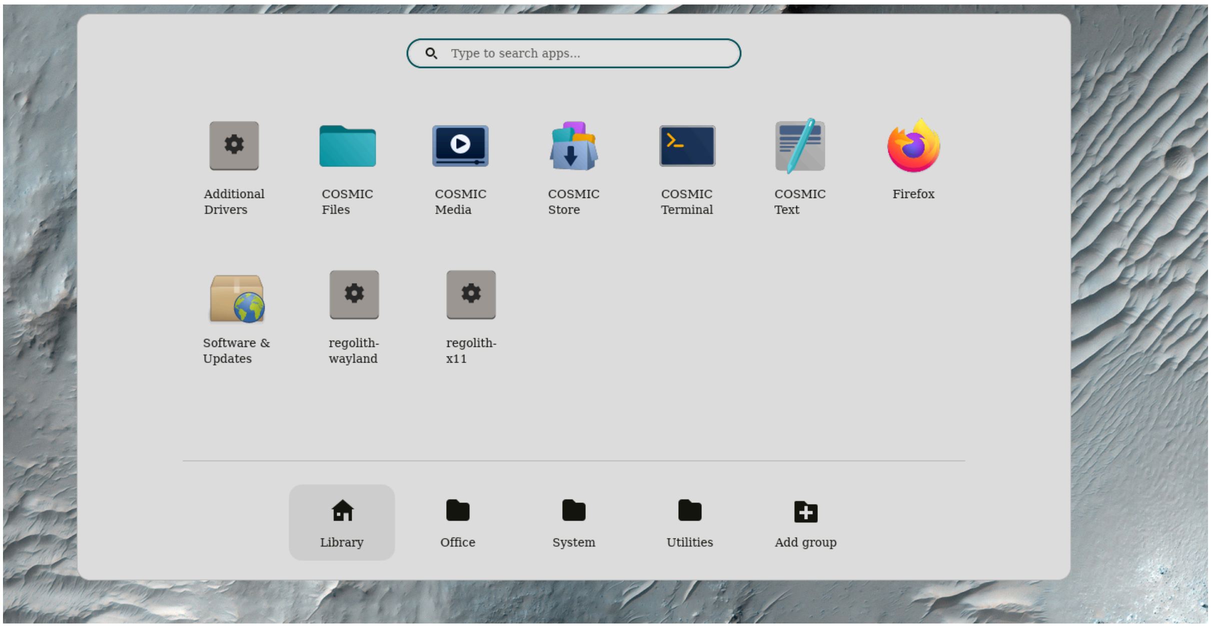
ilia → cosmic-launcher

Ilia currently offers dedicated pages for

1. Application Launcher
2. Terminal
3. Viewing Notifications via rofication
4. Viewing keybindings
5. windows and file tracker

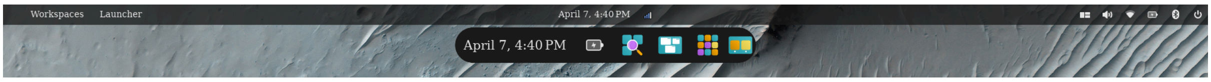
Though missing all these some people may prefer a more GUI interface instead. So most viable option is to provide an →

Option to change ilia with cosmic-launcher: on settings panel for super key action.



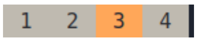
Cosmic-launcher: automatically adapts to settings changes done via cosmic-settings.

swaybar → cosmic-panel + dock



swaybar currently uses a i3status-rs backend for providing all the information about volume/wifi/disk-usage/memory-usage/volume/date-time and notification counts, including one applet (**nm-applet**). Whereas the cosmic-panel includes 7 more applets to present similar information.

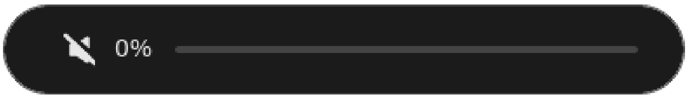
Suggested Changes:

1. All the applets should be fixed to respect the sway-workspaces drawing area.
2. Workspaces applet is to be fixed. (get workspaces names and count using sway-ipc and project it to current applet)
3. Addition of sway-bar style workspaces switcher 
4. Creation of a custom applet for **idle-inhibitor/caffeine** (The following button will not allow the system to go to sleep mode when enabled). The applet will open when clicked with the secondary button on the mouse that would enable further fine tuning of idle-inhibition settings.

regolith-osd → cosmic-osd (minor config change)

Currently OSD in regolith are defined in regolith-wm-config via a simple config definition in regolith-wm-config $wm.media.volume.step manages has a script that presents the OSD Graphics as soon as any XF86 media or volume keys are pressed.

Removing **mate-polkit-authentication-agent** from executing. The cosmic-osd directly integrates and works completely fine. Therefore we just need to remove the previous OSD overlays from regolith-wm-config.

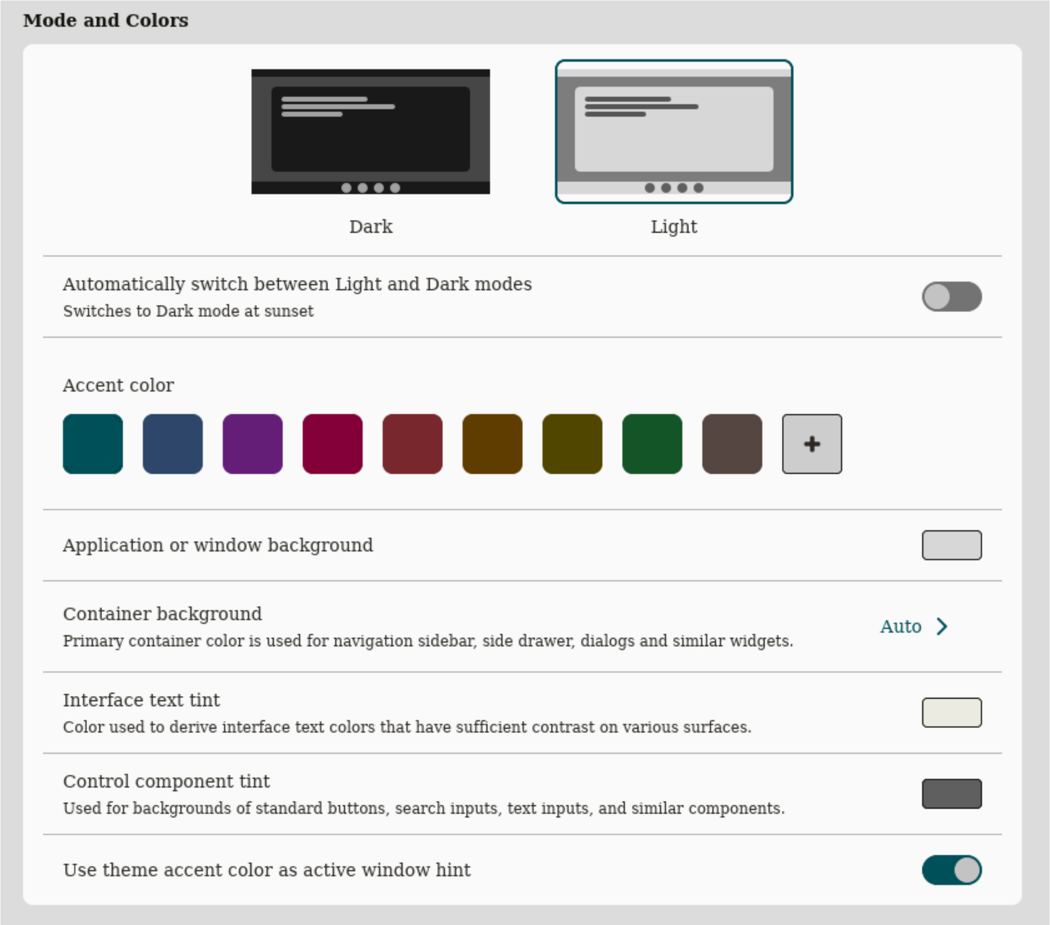


regolith-control-center → cosmic-settings

Fix Settings Pages

Desktop Page

1. Appearance Section



Mode and Colors

Dark

Light

Automatically switch between Light and Dark modes
Switches to Dark mode at sunset

Accent color

+

Application or window background

Container background
Primary container color is used for navigation sidebar, side drawer, dialogs and similar widgets.

Auto >

Interface text tint
Color used to derive interface text colors that have sufficient contrast on various surfaces.

Control component tint
Used for backgrounds of standard buttons, search inputs, text inputs, and similar components.

Use theme accent color as active window hint

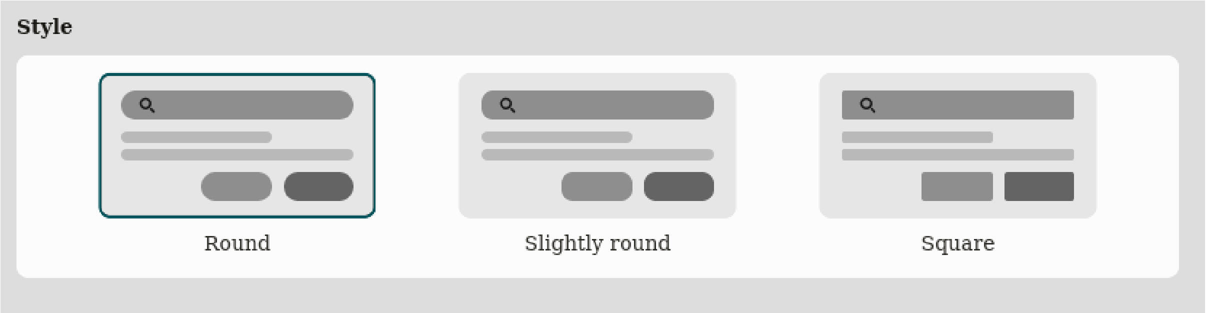
Buttons to Fix for Regolith:

1. Dark/Light Mode Buttons: should make regolith colors to adapt as well. (Additional Fix: They are not instantly responsive to change)
2. Accent color settings: should change the regolith theme.
3. Container Background + Control Component tint + Interface Text tint

Suggested Additions:

1. Regolith currently offers a set of predefined themes in the repository regolith-look-extra. Can define an option that uses the accent colors from those to directly use the previous infrastructure, also removing the need to **regolith-look-selector** binary.

2. Style Section



Style

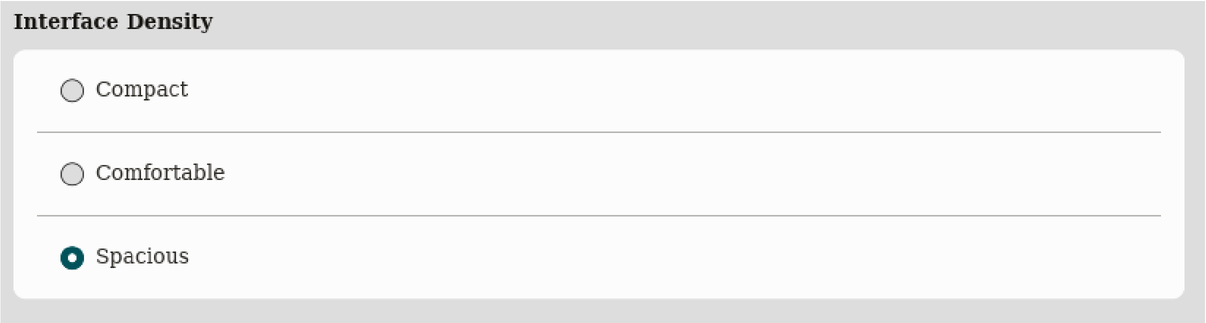
Round

Slightly round

Square

Should be completely removed: This cannot be applied for all the applications regolith currently comprise. The option does not show significant change when switched. (questionable though?)

3. Style Section + Window Management

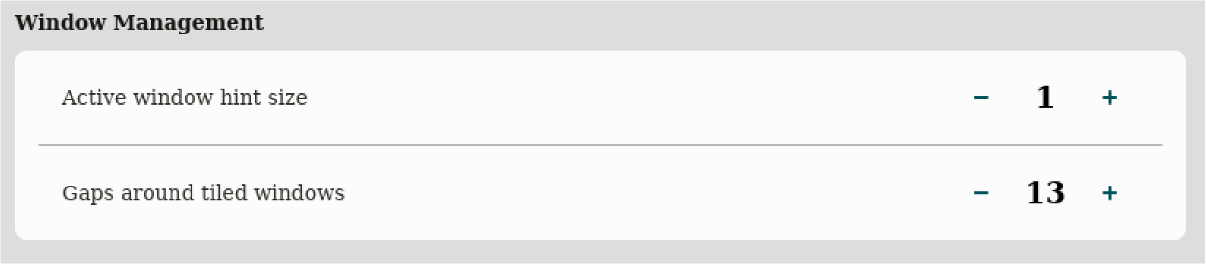


Interface Density

○ Compact

○ Comfortable

● Spacious



Window Management

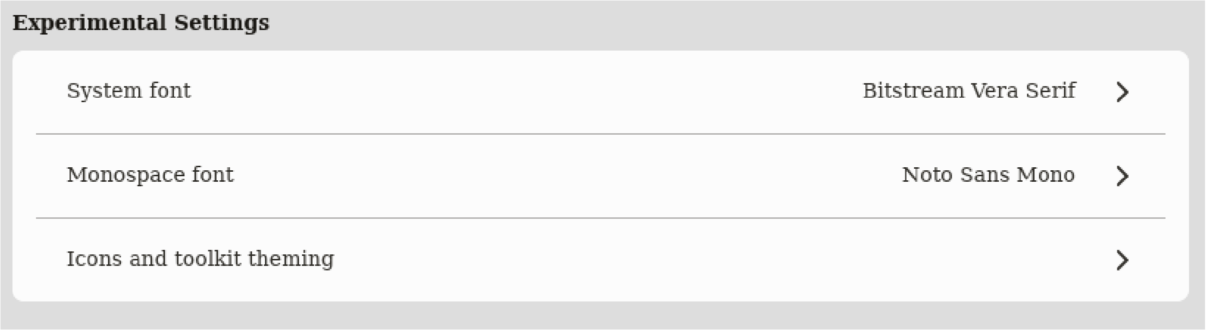
Active window hint size − 1 +

Gaps around tiled windows − 13 +

Can be fixed: By using `gaps inner <number>` settings using sway-ipc. Though already implemented in cosmic-ext-sway presented by Victoria.

Persistent to reboot: Since all the settings changed here would be managed by some program running in the background changing states using sway-ipc. Either killing the application of any glitches would revert back all settings to default regolith-config. Ensuring the following would require redesigning the current regolith's workflow or a helper application that reflect the changes to config respectively. (*Yet adding the program would even clutter everything more, this point needs some more concerned discussion.*)

4. Experimental Section



Experimental Settings

System font — Bitstream Vera Serif ›

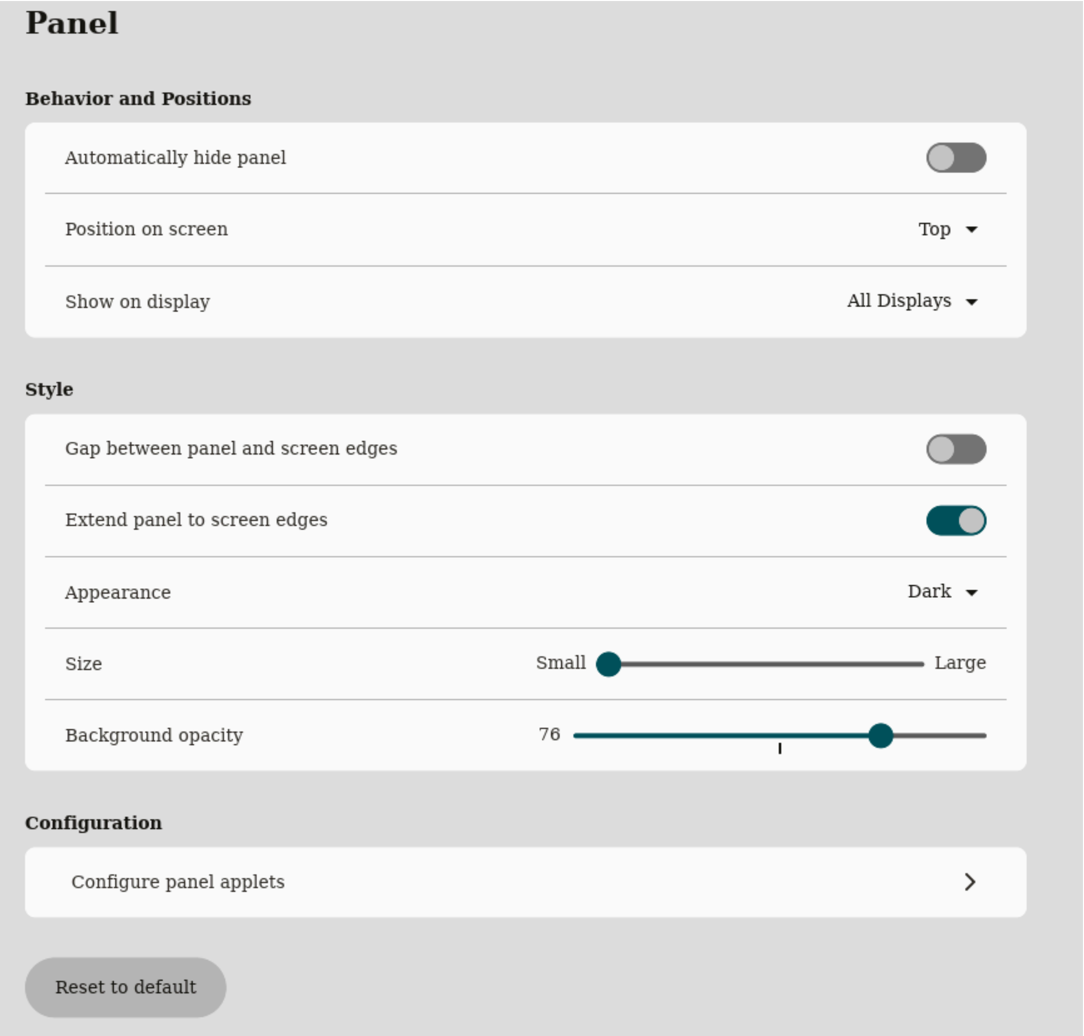
Monospace font — Noto Sans Mono ›

Icons and toolkit theming ›

Font Fixes: font-config for system-wide integration/ changing XDG variables should also achieve this (need more search for workarounds)

Icon and toolkit theming: (Need more searching for further comments on its functionality and configurability) Changing XDG Variables should do the job.

Panel Page



Panel

Behavior and Positions

Automatically hide panel

Position on screen — Top ▾

Show on display — All Displays ▾

Style

Gap between panel and screen edges

Extend panel to screen edges

Appearance — Dark ▾

Size — Small ● Large

Background opacity — 76

Configuration

Configure panel applets ›

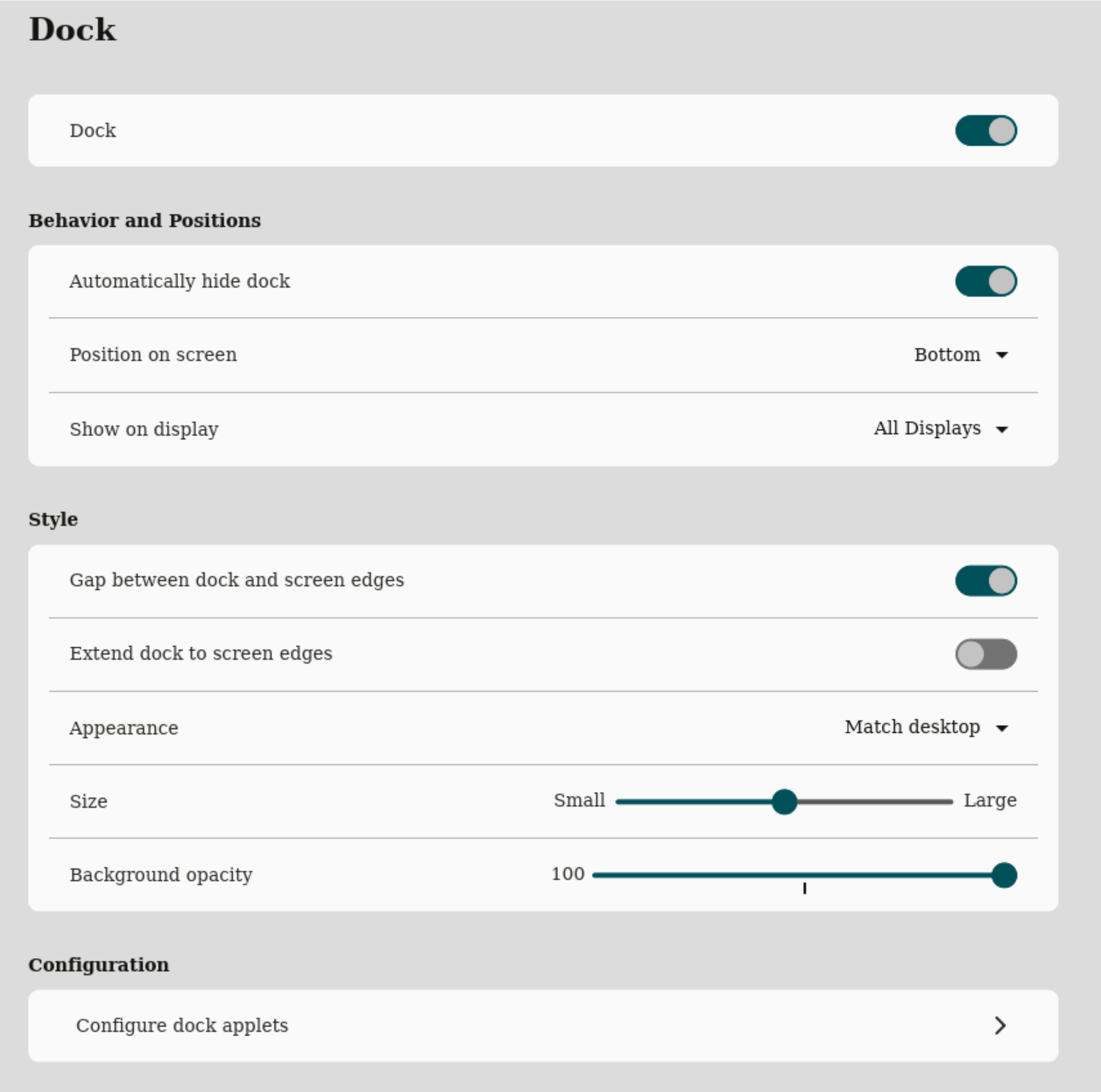
Reset to default

No Fixes Required: Currently is perfectly able to change panel settings on sway-regolith without any gluecode.

Suggested Additions:

1. Option to switch between swaybar/cosmic-panel. If swaybar is selected the following settings will be blanked accordingly.
2. Further additions to add custom options for swaybar customization as well.

Dock Page



Dock

Dock ●

Behavior and Positions

Automatically hide dock ●

Position on screen Bottom

Show on display All Displays

Style

Gap between dock and screen edges ●

Extend dock to screen edges ○

Appearance Match desktop

Size Small — Large

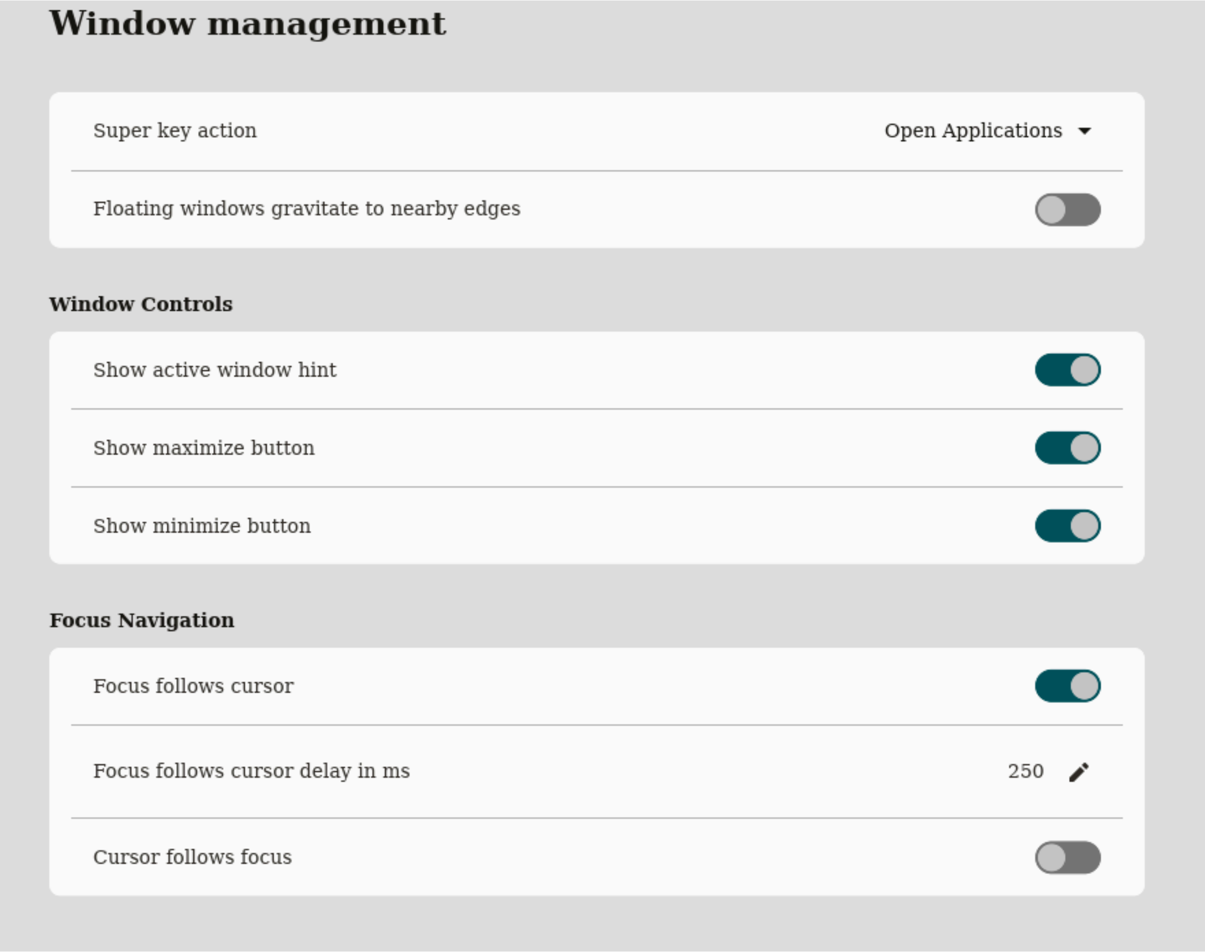
Background opacity 100

Configuration

Configure dock applets >

Mostly Works Fine: Every setting works perfectly similar to Panel Settings except the auto-hide button does not follow sway-regolith boundaries and if enabled the dock will always be visible.

Window Management Page



Window management

Super key action — Open Applications ▾

Floating windows gravitate to nearby edges ☐

Window Controls

Show active window hint ☑

Show maximize button ☑

Show minimize button ☑

Focus Navigation

Focus follows cursor ☑

Focus follows cursor delay in ms — 250 ✎

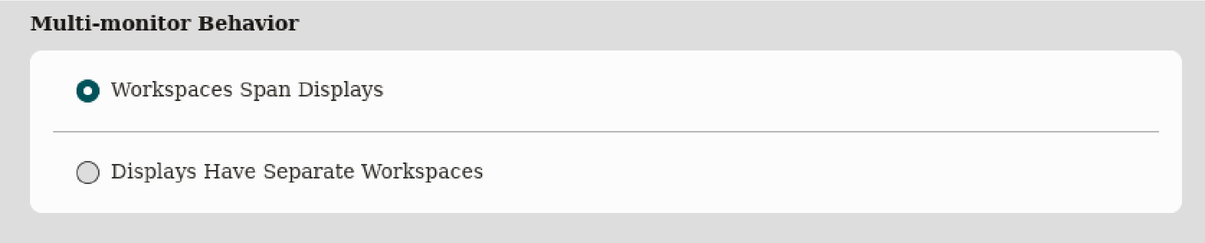
Cursor follows focus ☐

Super Key action: Can be settled using sway-ipc, meanwhile adding further options for opening ilia, opening launcher, and option to run a custom command instead using an input field.

Window Controls: No changes required..

Focus Navigation: The following can be implemented using the sway-ipc option corresponding to focus_follows_mouse. Focus follows cursor delays in ms may be implementable via a combination of swaymsg and libinput.

Workspaces Page

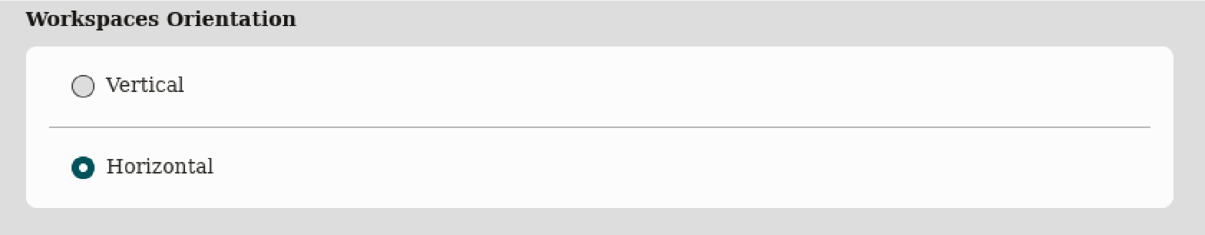


Multi-monitor Behavior

● Workspaces Span Displays

○ Displays Have Separate Workspaces

To be removed: Should be removed…



Workspaces Orientation

○ Vertical

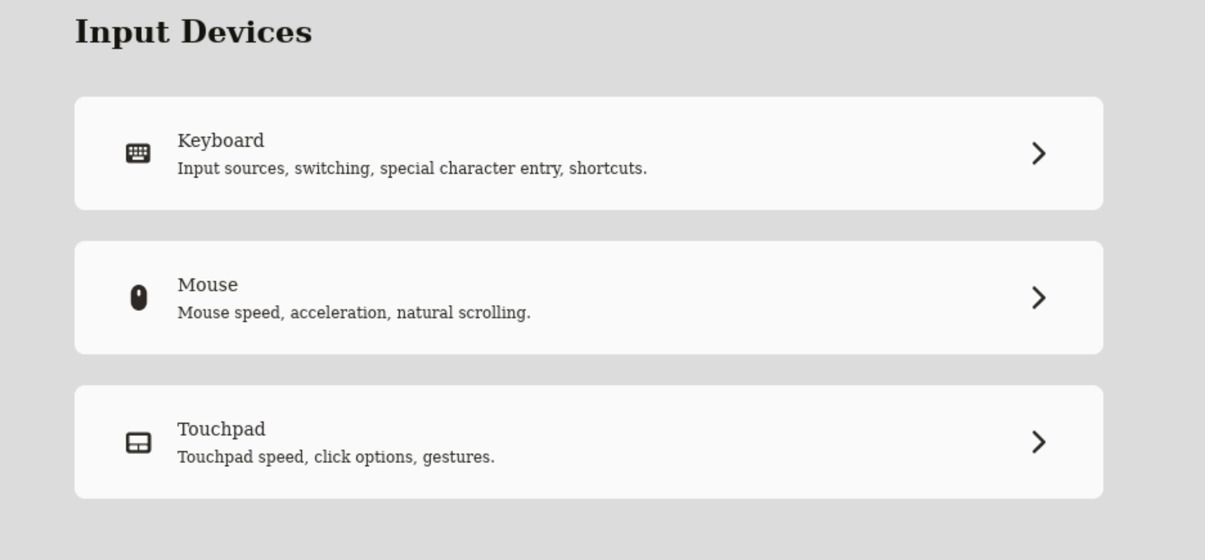
● Horizontal

To be removed: Again no such option provided by sway ...

Suggested Additions:

1. Option to set shortcut keys for workspace switch via an input field eg: <Super+Shift+1>
2. Option to set some specific workspace number to specific displays.
3. Option to set workspace names
4. Auto-remove/Allow empty workspaces option.

Input Settings Page



Input Devices

Keyboard
Input sources, switching, special character entry, shortcuts.

Mouse
Mouse speed, acceleration, natural scrolling.

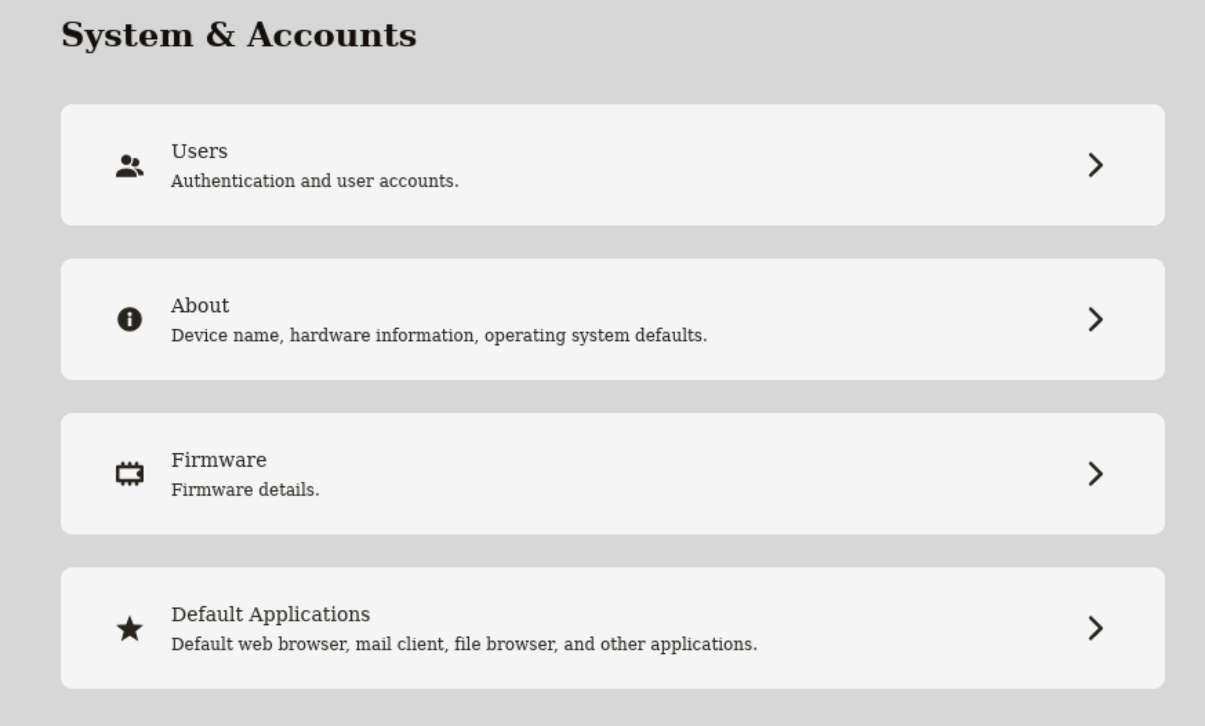
Touchpad
Touchpad speed, click options, gestures.

Keyboard, Mouse, Touchpad all of these settings are pseudo buttons currently. Probably missing a backend/daemon services (need more research)

Fix with no changes: After finding the reason and fixes or missing daemon, all these settings can be directly used as it is for regolith-wayland.

System & Accounts Page

About Section



System & Accounts

Users
Authentication and user accounts.

About
Device name, hardware information, operating system defaults.

Firmware
Firmware details.

Default Applications
Default web browser, mail client, file browser, and other applications.

Suggested Fixes:

1. The Firmware Details Page (currently-blank) would be fixed either by using tools like fwupdmgr,dmidecode with and pipe the results to the firmware section. OR a more relevant approach would be to use native rust application libraries like (heim, sysinfo) to connect these.
2. The default applications page contains pseudo option selectors for default applications. A simple program that edits ~/.config/mimeapps.list can be used to achieve this.

Add New Settings Pages

Regolith Configuration Section

I propose to add a custom regolith-config editing page that will edit the current user-config for regolith or at instances where required will use sway-ipc to enforce those settings. The following pages will be added under a regolith-config setting's section.

Decoration Settings Page

1. **Window Border Settings (Can be switched to appearance Page as well)**
   - Border Style → Dropdown (Options: none, normal, pixel)

- Border Width → Number Input with Slider (1-10px)
- Border Color → Color Picker
- Active Window Highlight → Toggle Switch

2. **Floating Window Settings**
   - Floating Border Style → Dropdown (Options: none, normal, pixel)
   - Floating Border Width → Number Input with Slider (1-5px)
   - Floating Border Color → Color Picker
   - Floating Maximum Size → Toggle Switch + Width/Height Inputs

3. **Gaps Configuration**
   - Enable Gaps → Toggle Switch
   - Inner Gap Size → Number Input with Slider (0-50px)
   - Outer Gap Size → Number Input with Slider (0-50px)
   - Smart Gaps → Toggle Switch
   - Per-Workspace Gaps → Toggle Switch + Workspace Selector

4. **Hide Borders Options**
   - Edge Border Hiding → Dropdown (Options: none, vertical, horizontal, both, smart)
   - Hide When Single Window → Toggle Switch

5. **Window Title Options**
   - Show Window Titles → Toggle Switch
   - Title Font → Font Selector
   - Title Alignment → Radio Buttons (Left, Center, Right)
   - Title Padding → Number Input (0-20px)

6. **Apply Changes** → Primary Button
7. **Reset to Defaults** → Secondary Button

General Settings Page

1. **Focus Behavior**
   - Focus Follows Mouse → Dropdown (Options: no, yes, always)
   - Focus Wrapping → Dropdown (Options: no, yes, force, workspace)
   - Mouse Warping → Dropdown (Options: none, output, container)

2. **Workspace Behavior**
   - Default Workspace Layout → Dropdown (Options: default, stacking, tabbed)
   - Auto Back and Forth → Toggle Switch
   - Workspace Auto-Assignment → Toggle Switch + App-Workspace Editor

3. **Display & Outputs**

- Monitor Configuration → Button (Opens visual layout editor)
- Clamshell Mode → Dropdown (Options: do nothing, dock, undock)
- Workspace Output Assignments → Multi-line Editor

4. **Application Startup**
   - Auto-start Applications → List View with Add/Remove buttons
   - Default Applications → Button (Opens application chooser)

5. **Bar Settings**
   - Bar Position → Dropdown (Options: top, bottom, none)
   - Bar Mode → Dropdown (Options: dock, hide, invisible)
   - Bar Font → Font Selector
   - Status Bar Config → File Path Selector with Edit Button
   - Workspace Min Width → Number Input
   - Show Workspace Numbers → Toggle Switch

6. **Apply Changes** → Primary Button
7. **Reset to Defaults** → Secondary Button

Keybindings Settings Page

1. **Modifier Key**
   - Primary Modifier ($mod) → Dropdown (Options: Win/Super, Alt, Ctrl)
   - Secondary Modifier ($alt) → Dropdown (Options: Win/Super, Alt, Ctrl)

2. **Window Management**
   - Focus Navigation (up/down/left/right) → Keybinding Inputs
   - Move Window (up/down/left/right) → Keybinding Inputs
   - Resize Mode Toggle → Keybinding Input
   - Fullscreen Toggle → Keybinding Input
   - Floating Toggle → Keybinding Input
   - Split Orientation → Keybinding Inputs
   - Layout Toggle → Keybinding Inputs

3. **Workspace Management**
   - Switch to Workspace → Keybinding Inputs (1-10)
   - Move Window to Workspace → Keybinding Inputs (1-10)
   - Next Free Workspace → Keybinding Input
   - Move to Next Free Workspace → Keybinding Input
   - Carry to Next Free Workspace → Keybinding Input

4. **Launchers**
   - Application Launcher → Keybinding Input + Command Input
   - Command Launcher → Keybinding Input + Command Input

   - Window Switcher → Keybinding Input + Command Input
   - File Search → Keybinding Input + Command Input
   - Look Selector → Keybinding Input + Command Input
5. **Session Controls**
   - Lock Screen → Keybinding Input + Command Input
   - Exit App → Keybinding Input
   - Terminate App → Keybinding Input
   - Reload Config → Keybinding Input
   - Logout → Keybinding Input + Command Input
   - Reboot → Keybinding Input + Command Input
   - Shutdown → Keybinding Input + Command Input
   - Sleep → Keybinding Input + Command Input

6. **Media & System Controls**
   - Volume Controls → Keybinding Inputs + Command Inputs
   - Brightness Controls → Keybinding Inputs + Command Inputs
   - Media Playback → Keybinding Inputs + Command Inputs
   - Screenshot Functions → Keybinding Inputs + Command Inputs

7. **Import/Export** → Button Group (Import from file, Export to file, Reset all)
8. **Apply Changes** → Primary Button
9. **Reset to Defaults** → Secondary Button

Variables Settings Page

1. **Resource Variables**
   - Filter Variables → Search Input
   - Variable List → Table View (Name, Value, Edit Button)
   - Custom Variables → Add/Remove Buttons + Name/Value Inputs
2. **Path Settings**
   - Status Command → Text Input + File Browse Button
   - Status Config → Text Input + File Browse Button
   - Lock Command → Text Input + File Browse Button
   - Logout Command → Text Input + File Browse Button
   - Reboot Command → Text Input + File Browse Button
   - Shutdown Command → Text Input + File Browse Button
3. **Customization Variables**
   - Ilia Stylesheet → Text Input + File Browse Button
   - Bar Separator Character → Text Input
   - Media Volume Step → Number Input (1-10)
   - Brightness Step → Number Input (1-10)
   - Gap Increment Size → Number Input (1-20)
4. **Application Commands**
   - Settings Application → Text Input (regolith-control-center or gnome-control-center)

- Display Settings → Text Input
- WiFi Settings → Text Input
- Bluetooth Settings → Text Input
- File Browser → Text Input
- Terminal → Text Input

5. **Reset to System Default** → Button (per variable)
6. **Apply Changes** → Primary Button
7. **Reset All to Defaults** → Secondary Button

Misc Settings Page

1. **Input Devices**
    - Keyboard Layouts → Multi-select list with ordering + Add/Remove Buttons
    - Layout Switching → Keybinding Inputs
    - Touchpad Gestures → Table View (Gesture, Action, Edit Button)
2. **Services Management**
    - D-Bus Activation → Toggle Switch
    - XDG Desktop Portal → Toggle Switch
    - Polkit Method → Dropdown (Options: mate-polkit, mate-polkit with x11)
    - Audio Idle Inhibit → Toggle Switch
    - Avizo Notifications → Toggle Switch
    - Playerctl Service → Toggle Switch
3. **Screenshot Settings**
    - Screenshot Directory → Directory Picker + Text Input
    - Screenshot Quality → Number Input with Slider (1-100)
    - Screenshot Format → Dropdown (Options: PNG, JPG)
    - Include Cursor → Toggle Switch
    - Notification → Toggle Switch
4. **Window Rules**
    - App-specific Rules → Table View (Window Class, Properties, Edit Button)
    - Add New Rule → Button (Opens rule editor dialog)
    - Import Rules → Button (Import from file)
5. **Advanced Options**
    - Debug Log Level → Dropdown (Options: quiet, verbose, debug)
    - Configuration Directory → Directory Picker + Text Input
    - Save On Exit → Toggle Switch

6. **Apply Changes** → Primary Button
7. **Reset to Defaults** → Secondary Button

Timeline

The learning phase will be completed in the time provided for Application Review Period + Community Bonding Period combined. I have tried to add NixOS integrations here as a part because I daily drive NixOS, and would try to make everything Nix Compatible from the start (though the cosmic-epoch already support NixOS quite perfectly), I will be making additions on regolith-nix mostly in this purpose and during my final testing I will overview all the things in Debian. Ensuring the following project serves both sides equally.

Learning Phase (3 Weeks)

Week 1-2: Time for learning iced-rs (the framework used by cosmic to develop its GUI applications). Getting familiar with iced-rs by developing custom setting's pages/tweaking applications and playing around with it.

Week 2-3: Learn about and set up debugging tools (Wayland logs, sway IPC hooks, journalctl filters). Also, will try to set up a fast paced development environment with bash-scripts to fasten the development via these components. Will test the possibility of integration/dev setup on NixOS as well.

Phase 1: Settings Integration and UI Improvements

Weeks 3-5: Settings Pages

- Integrate cosmic-settings Display page functionality
    - Ensure display settings persist across sessions
    - Test multi-monitor configurations
- Implement cosmic-settings Panel/Dock functionality
    - Add option to switch between swaybar/cosmic-panel
    - Fix auto-hide functionality for dock in sway environment
    - Add custom applet for idle-inhibitor
- Begin work on Desktop section settings integration
    - Implement gaps settings via sway-ipc
    - Connect accent color settings to regolith theme system

Weeks 5-7: Extended Settings Integration

- Create custom Regolith Configuration section in cosmic-settings
    - Implement window border settings page
    - Add floating window configuration options
    - Build gaps configuration interface
- Implement Input Settings page functionality
    - Research and identify missing daemon dependencies

    - Integrate keyboard/mouse/touchpad settings with sway
- Add About section firmware details
    - Implement system information retrieval using rust libraries (heim/sysinfo)
    - Connect default applications page to mimeapps.list

Phase 2: Custom Pages + NixOS Compatibility(5 weeks)

Weeks 7-9: Custom Settings Pages Implementation

- Develop General Settings page
    - Implement focus behavior options/cursor interaction options
    - Add workspace behavior configuration + settings startups applications
    - Create display & output management interface via sway config
- Create Key Bindings Settings page
    - Build modifier key selection interface + Modifier Behaviour Command Input Buttons
    - Implement keybinding editors for window/workspace management
    - Add media & system control configuration

Weeks 9-11: Some Viable Integrations and Refinements

- Implement Variables and Misc Settings pages
    - Create resource variables configuration interface
    - Implementing input devices configuration options (requirement is yet questionable though)
- Connect cosmic-notifications with sway (if feasible)
    - Researching alternative implementation approaches if custom protocols are blocking
    - Implement compatibility layer if possible & if not will change the notification applet to support regolith-rofication.
- Complete workspaces integration in cosmic-panel
    - Finalize sway workspaces detection and interaction
    - Making sure the existing workspaces applet works properly with sway.
    - Add custom workspace switcher buttons on bar as mentioned above

Week 11-12: NixOS Integration and Module Development

- **Component Integration**
    - Will Integrate with existing Regolith Nix packages you've already contributed to nixpkgs
    - Create a unified Regolith-Cosmic FHS meta-package that pulls in all required custom components

- **NixOS Module Creation**
  - Develop comprehensive NixOS module for Regolith-Cosmic with appropriate options structure for enabling regolith with cosmic
  - Implement service definitions for all background daemons (cosmic-settings-daemon, etc.)

Phase 3: Testing, Documentation (2-3 weeks)

Focus: Ensuring stability and preparing for release

Weeks 12-13: Testing and Bug Fixing

- Comprehensive testing across different hardware configurations
- WILL Fix identified issues and edge cases
- Performance optimization and memory usage testing (currently each application approximately takes 20MB of memory space when running)
- Create user migration tools if needed
- **And the most important part:** Thinking of a name for this new regolith flavour :) (**Cosmolith** would be quite good right?)

Week 13-14 (if needed): Documentation and Release Preparation

- Complete user and developer documentation
- Create onboarding guides for new Regolith-Cosmic users
- Prepare release packages and announcements

Factor Influencing Timeline

1. **Configuration Persistence**: Ensuring settings persist across reboots might require redesigning Regolith's current workflow or implementing helper applications, which could extend the timeline.

2. **Upstream Changes**: The cosmic-epoch project is still in alpha, so significant upstream changes could require adaptation of the integration work.

Additional Information about the Timeline

- The timeline mentioned above is subject to change and is only an approximate outline of my project work. I will stick to or exceed this schedule and create a more detailed schedule during the pre-GSoC and community bonding phase.

- I've no other commitments during the summer and can dedicate 30 to 35 hours a week. During the last month of the project, my college will begin, and I'll be able to commit a max of 20 a week. Due to the same, I will do a significant portion of the work before this period.

- Time will be divided (according to workload) each week amongst planning, learning, coding, documenting and testing features. All documentation will go hand in hand with the development.

Commitments

Commitments During weekdays (Monday through Friday), I will be able to dedicate 3-5 hours to project work, ensuring alignment with Indian working hours. This allows for efficient communication and collaboration. To compensate for the reduced weekday schedule, I'm happy to invest additional hours on weekends (Saturday and Sunday) – typically 7-8 hours per day. This extended weekend commitment ensures that I meet the proposed timeline..

POST GSOC Plans

Wallpaper Based Theming in Regolith: Many modern ricers and even Android these day support wallpaper based theming either via Material U themes or using applications like pywal/wallust to generate color schemes and using those to generate colors schemes referencing this base16 scheme to further application’s css/ config to have a neat UI (I do the same using both of these). Post GSoC just as a fun project I would like to implement this natively to regolith with a setting’s button to enable or disable this. The main benefit of such a theming system is all applications (firefox, vscode, obsidian, GTK themes, sway, spotify) natively or externally support pywal theming to maintain consistency in the system’s appearance.

For this either we can use already existing applications like wallust which is completely written in rust and its core library can be used to implement the same functionally or we can use some glue code (which at this point I think should be avoided).

THANK YOU