

# Pull Request: Linear-grade Motion (Framer) + Foundational Component Kit

Branch: `feat/ui-v2-motion-components`

**Scope:** Add **Framer Motion** for tasteful page/element transitions (reduced-motion aware) and ship a **foundational component kit** (inputs, select, switch, dropdown, modal, tabs, tooltip, toast, badge, card) consistent with the existing **styled-components** design system.

Aligns with canvases:

- *YooHoo Guru — Design System (Linear-grade) + UI PR v1*
- *PR – Welcome + SkillShare & Angel's List (v1)*

---

## 1) What users will see (high level)

- Smooth **page transitions** (fade+lift 8–12px) when navigating between Home, Angel's List, and Marketplace.
- **Hero/tiles** subtly animate in (staggered, 60–80ms) on first paint.
- Tooltips, modals, dropdowns, and toasts feel **fast and precise**; no cartoon motion.
- Forms look and behave consistently: floating labels, clear validation, accessible focus.

---

## 2) Dependencies

```
# frontend
npm i framer-motion
```

No other libs added. Motion integrates with current `styled-components` theme.

---

## 3) Motion architecture

### 3.1 Tokens (reuse theme values)

We reuse `theme.motion` and durations from v1. Add spring presets.

..

```

const motion = {
  duration: { fast: 0.14, med: 0.2, slow: 0.26 },
  easeIn: [0.2, 0.7, 0.25, 1],
  easeOut: [0.3, 0.1, 0.2, 1],
  springSoft: { type: 'spring', stiffness: 340, damping: 30, mass: 0.9 },
  springTight: { type: 'spring', stiffness: 480, damping: 38, mass: 0.9 }
};
export default motion;

```

### 3.2 Reduced motion hook

..

```

import { useEffect, useState } from 'react';
export default function useReducedMotion() {
  const [prefers, setPrefers] = useState(false);
  useEffect(() => {
    const mq = window.matchMedia('(prefers-reduced-motion: reduce)');
    const onChange = () => setPrefers(mq.matches);
    onChange();
    mq.addEventListener?.('change', onChange);
    return () => mq.removeEventListener?.('change', onChange);
  }, []);
  return prefers;
}

```

### 3.3 Route transitions wrapper

..

```

import React from 'react';
import { AnimatePresence, motion } from 'framer-motion';
import useReducedMotion from '../../hooks/useReducedMotion';

const variants = {
  initial: { opacity: 0, y: 8 },
  enter: { opacity: 1, y: 0 },
  exit: { opacity: 0, y: -8 }
};

export default function RouteTransition({ locationKey, children }) {
  const reduced = useReducedMotion();
  if (reduced) return <>{children}</>;
  return (

```

```

    <AnimatePresence mode="wait">
      <motion.div key={locationKey}
        initial="initial" animate="enter" exit="exit"
        variants={variants}
        transition={{ duration: 0.20, ease: [0.2,0.7,0.25,1] }}
        style={{ willChange: 'transform, opacity' }}
      >
        {children}
      </motion.div>
    </AnimatePresence>
  );
}

```

Usage (wrap your `Routes` outlet). If using React Router in `Layout.js`:

```

import { useLocation, Outlet } from 'react-router-dom';
import RouteTransition from './motion/RouteTransition';

export default function Layout() {
  const { key } = useLocation();
  return (
    <main>
      <RouteTransition locationKey={key}>
        <Outlet />
      </RouteTransition>
    </main>
  );
}

```

### 3.4 Element entrance utilities

..

```

import { motion } from 'framer-motion';
import useReducedMotion from '../../hooks/useReducedMotion';

export function FadeIn({ delay = 0, y = 10, children, ...rest }) {
  const reduced = useReducedMotion();
  if (reduced) return <div {...rest}>{children}</div>;
  return (
    <motion.div
      initial={{ opacity: 0, y }}
      animate={{ opacity: 1, y: 0 }}
      transition={{ duration: 0.22, delay, ease: [0.2,0.7,0.25,1] }}
      {...rest}
    >
      {children}
    </motion.div>
  );
}

```

```

    >{children}</motion.div>
  );
}

```

In `HomePage`, wrap each tile: `<FadeIn delay={i*0.06}><Tile>...</Tile></FadeIn>`.

## 4) Component kit (styled-components)

All components: AA contrast, focus-visible outlines, keyboard support.

### 4.1 TextField (floating label)

..

```

import React from 'react';
import styled, { css } from 'styled-components';

const Wrap = styled.label`
  display:block; position:relative; margin: 0 0 16px;
`;
const Input = styled.input`
  width:100%; padding: 14px 12px 12px; color: var(--text);
  background: var(--surface); border:1px solid var(--border); border-radius:
var(--r-md);
  transition: border-color var(--t-fast) ease;
  &:focus{ outline: none; border-color:#2E3540; }
  &:disabled{ opacity:.6; cursor:not-allowed; }
`;
const Label = styled.span`
  position:absolute; left:12px; top:12px; color: var(--muted); pointer-
events:none;
  transition: transform var(--t-fast) ease, font-size var(--t-fast) ease, top
var(--t-fast) ease;
  ${Input}:focus ~ &,
  ${Input}:not(:placeholder-shown) ~ &{
    top:6px; transform: translateY(-10px); font-size:12px; color:#AEB8C4;
  }
`;
const Hint = styled.div` margin-top:6px; color: var(--muted); font-size: 12px;
`;
const Error = styled.div` margin-top:6px; color: var(--err); font-size: 12px; `;

export default function TextField({ label, hint, error, ...props }){
  return (
    <Wrap>

```

```

      <Input placeholder=" " aria-invalid={!!error} {...props} />
      <Label>{label}</Label>
      {error ? <Error role="alert">{error}</Error> : hint ? <Hint>{hint}</
Hint> : null}
    </Wrap>
  );
}

```

## 4.2 Select (Listbox)

..

```

import React, { useState, useRef, useEffect } from 'react';
import styled from 'styled-components';

const Box = styled.button`
  width:100%; text-align:left; padding:12px; border:1px solid var(--border);
  background: var(--surface); border-radius: var(--r-md); color: var(--text);
`;

const List = styled.ul`
  position:absolute; z-index:40; margin-top:6px; width:100%; list-style:none;
padding:6px;
  background: var(--elev); border:1px solid var(--border); border-radius: var(--
r-md);
`;

export default function Select({ value, onChange, options = [],
placeholder='Select...' }){
  const [open, setOpen] = useState(false);
  const ref = useRef(null);
  useEffect(() => {
    const onDoc = (e) => { if (open && ref.current && !
ref.current.contains(e.target)) setOpen(false); };
    document.addEventListener('mousedown', onDoc);
    return () => document.removeEventListener('mousedown', onDoc);
  }, [open]);
  return (
    <div ref={ref} style={{ position:'relative' }}>
      <Box aria-haspopup="listbox" aria-expanded={open} onClick={() =>
setOpen(v=>!v)}>
        {value ? options.find(o=>o.value===value)?.label : placeholder}
      </Box>
      {open && (
        <List role="listbox">
          {options.map(opt => (
            <li key={opt.value} role="option" aria-selected={opt.value===value}

```

```

        onClick={() => { onChange?.(opt.value); setOpen(false); }}
        style={{ padding: '8px 10px', borderRadius: 8, cursor: 'pointer',
background: opt.value === value ? 'rgba(124,140,255,.10)' : 'transparent' }}>
        {opt.label}
      </li>
    )}
  </List>
)
</div>
);
}

```

### 4.3 Switch

..

```

import React from 'react';
import styled from 'styled-components';

const Track = styled.button`
  width: 44px; height: 26px; border-radius: 999px; position: relative;
  background: ${({ checked }) => checked ? 'rgba(124,140,255,.22)' : 'var(--surface)'};
  border: 1px solid var(--border); transition: background var(--t-fast) ease;
`;
const Thumb = styled.span`
  position: absolute; top: 2px; left: ${({ checked }) => checked ? '22px' : '2px'};
  width: 22px; height: 22px; border-radius: 999px; background: #DDE2FF;
  transition: left var(--t-fast) ease;
`;
export default function Switch({ checked, onChange, ...rest }) {
  return (
    <Track role="switch" aria-checked={checked} onClick={() => onChange?.(!checked)} checked={checked} {...rest}>
      <Thumb checked={checked} />
    </Track>
  );
}

```

### 4.4 Dropdown Menu

..

```

import React, { useRef, useState, useEffect } from 'react';
import styled from 'styled-components';

```

```

const Btn =
  styled.button` padding:.5rem .75rem; border:1px solid var(--border); background:
  var(--surface); border-radius: var(--r-sm); `;
const Menu = styled.div` position:absolute; z-index:50; margin-top:6px; min-
width:180px; background: var(--elev); border:1px solid var(--border); border-
radius: var(--r-md); padding:6px; `;

export default function Dropdown({ button, items=[] }){
  const [open, setOpen] = useState(false);
  const ref = useRef(null);
  useEffect(() => {
    const onDoc = (e) => { if (open && ref.current && !
ref.current.contains(e.target)) setOpen(false); };
    document.addEventListener('mousedown', onDoc);
    return () => document.removeEventListener('mousedown', onDoc);
  }, [open]);
  return (
    <div ref={ref} style={{ position:'relative', display:'inline-block' }}>
      <Btn onClick={() => setOpen(v=>!v)} aria-haspopup="menu" aria-
expanded={open}>{button}</Btn>
      {open && (
        <Menu role="menu">
          {items.map((it, i) => (
            <button key={i} role="menuitem" onClick={() => { it.onSelect?();
setOpen(false); }}
              style={{ display:'block', width:'100%', textAlign:'left',
padding:'8px 10px', borderRadius:8, background:'transparent' }}>
                {it.label}
              </button>
            ))}
          </Menu>
        )}
    </div>
  );
}

```

## 4.5 Modal/Dialog

..

```

import React, { useEffect } from 'react';
import styled from 'styled-components';

const Backdrop = styled.div`
  position:fixed; inset:0; background: rgba(0,0,0,.45); display:flex; align-

```

```

items:center; justify-content:center; z-index:60;
`;
const Sheet = styled.div`
  width:min(640px, 92vw); background: var(--elev); border:1px solid var(--
border); border-radius: var(--r-lg); padding: 20px;
`;
export default function Modal({ open, onClose, children }){
  useEffect(() => {
    const onKey = (e) => { if (e.key === 'Escape' && open) onClose?.(); };
    document.addEventListener('keydown', onKey);
    return () => document.removeEventListener('keydown', onKey);
  }, [open, onClose]);
  if (!open) return null;
  return (
    <Backdrop onClick={onClose}>
      <Sheet role="dialog" aria-modal={true} onClick={(e) =>
e.stopPropagation()}>
        {children}
      </Sheet>
    </Backdrop>
  );
}

```

## 4.6 Tabs

..

```

import React from 'react';
import styled from 'styled-components';

const Bar = styled.div` display:flex; gap:6px; border-bottom: 1px solid var(--
border); `;
const Tab = styled.button`
  padding:.5rem .75rem; border-radius: var(--r-sm) var(--r-sm) 0 0; border:1px
solid transparent;
  &.active{ border-color: var(--border); border-bottom-color: var(--elev);
background: var(--surface); }
`;
export default function Tabs({ value, onChange, items=[] }){
  return (
    <>
      <Bar>
        {items.map(it => (
          <Tab key={it.value} className={value===it.value ? 'active' : ''}
onClick={() => onChange?.(it.value)}>
            {it.label}
          </Tab>
        ))}
      </Bar>
    </>
  );
}

```



```

        </Tab>
      )})
    </Bar>
    {items.find(i=>i.value===value)?.children}
  </>
);
}

```

## 4.7 Tooltip

..

```

import React, { useState } from 'react';
import styled from 'styled-components';

const Wrap = styled.span` position:relative; display:inline-block; `;
const Bubble = styled.span`
  position:absolute; bottom: calc(100% + 8px); left: 50%; transform:
  translateX(-50%);
  background: #0D1117; color:#E8EDF2; border:1px solid #30363D; border-radius:
  8px; padding:6px 8px; white-space:nowrap; z-index:70;
`;
export default function Tooltip({ content, children }){
  const [open, setOpen] = useState(false);
  return (
    <Wrap onMouseEnter={() => setOpen(true)} onMouseLeave={() => setOpen(false)}
    >
      {children}
      {open && <Bubble role="tooltip">{content}</Bubble>}
    </Wrap>
  );
}

```

## 4.8 Toast (portal)

..

```

import React, { useEffect, useState } from 'react';
import ReactDOM from 'react-dom';

function Portal({ children }){
  const [el] = useState(() => document.createElement('div'));
  useEffect(() => { document.body.appendChild(el); return () => el.remove(); },
  [el]);
  return ReactDOM.createPortal(children, el);
}

```

```

}

export function useToast() {
  const [items, setItems] = useState([]);
  const notify = (msg, ttl=3000) => {
    const id = Math.random().toString(36).slice(2);
    setItems(arr => [...arr, { id, msg }]);
    setTimeout(() => setItems(arr => arr.filter(x => x.id !== id)), ttl);
  };
  const ToastHost = () => (
    <Portal>
      <div style={{ position:'fixed', right:16, bottom:16, display:'grid', gap:
8 }}>
        {items.map(t => (
          <div key={t.id} style={{ background:'var(--elev)', border:'1px solid
var(--border)', borderRadius:12, padding:'10px 12px' }}>{t.msg}</div>
        ))}
      </div>
    </Portal>
  );
  return { notify, ToastHost };
}

```

## 4.9 Badge & Card

..

```

import styled from 'styled-components';
export default styled.span`
  display:inline-flex; align-items:center; gap:6px;
  padding:2px 8px; border-radius: 999px; border:1px solid var(--border);
  background: rgba(124,140,255,.10); color:#DDE2FF; font-size:12px;
`;

```

..

```

import styled from 'styled-components';
export default styled.div`
  background: var(--surface); border:1px solid var(--border); border-radius:
var(--r-lg); box-shadow: var(--shadow-card);
`;

```

## 5) File-level changes summary

- **Add** `theme/motion.js`
  - **Add** `hooks/useReducedMotion.js`
  - **Add** `components/motion/RouteTransition.js`, `components/motion/FadeIn.js`
  - **Wrap** routes outlet in `RouteTransition` (Layout)
  - **Add** components in `components/ui/...` as above
  - **Update** Home tiles to use `<FadeIn delay={...}>`
  - **(Optional)** Apply `<FadeIn>` to Angel's List cards
- 

## 6) Usage examples

Home tiles:

```
{tiles.map((t, i) => (  
  <FadeIn key={t.key} delay={i*0.06}>  
    <Tile>{t.content}</Tile>  
  </FadeIn>  
))}
```

Dropdown in Header:

```
<Dropdown button={<span>Account ▾</span>} items=[  
  { label: 'Profile', onSelect: () => navigate('/profile') },  
  { label: 'Sign out', onSelect: logout }  
] />
```

Toasts:

```
const { notify, ToastHost } = useToast();  
// in root layout  
<ToastHost />  
// on some action  
notify('Saved');
```

---

## 7) Acceptance criteria

-

---

## 8) A11y & perf notes

- Reduce motion honored; no animation in that mode.
  - All interactive elements are buttons/links with focus styles.
  - Keep animation layers simple; `will-change` only where needed.
  - Lazy-mount heavy menus/modals.
- 

## 9) Risks & mitigations

- **Focus trap completeness:** current modal closes on ESC and blocks backdrop click. Full tab trapping to be added using `tabindex` sentinels in v2.1.
  - **Select edge cases:** for very long lists, prefer a searchable command palette (future component) or native `<select>` on mobile.
- 

## 10) Follow-ups

- v2.1: Full focus trap + aria attributes audit; keyboard nav for listbox with arrow keys and typeahead.
- v2.2: Command Palette (⌘K), segmented control, breadcrumb, pagination, data table.
- v2.3: Framer layout animations for reordering lists (e.g., sessions, requests).