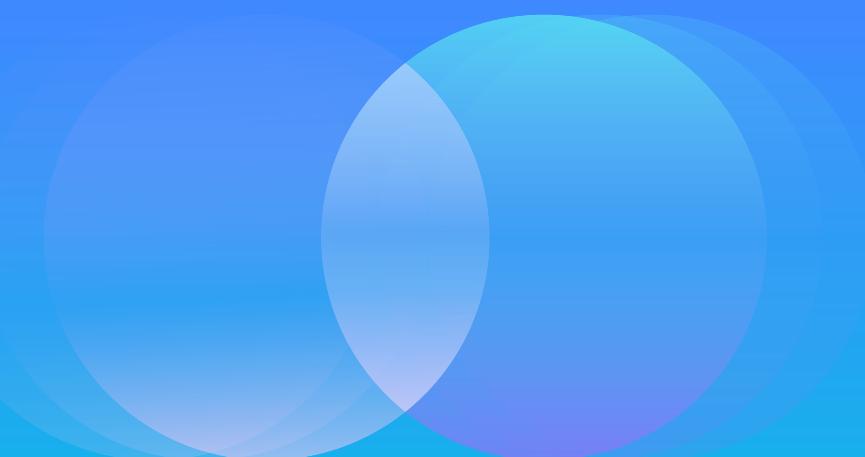


product design system from
planning to interaction



- Research
- UI Design & UX Modeling
- Screen Development



jajac, Product Design System

‘자작(jajac)’은 서비스를 만들어 내는 버치의 디자인 시스템으로, Atomic Design Methodology를 기반으로 고객에게 새로운 가치를 제공하는 것을 목표로 하고 있습니다.



Library System

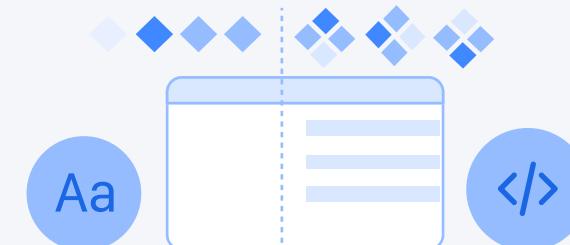
jajac, Strategic Approach for Efficiency

Research, UI Design & UX Modeling, Screen Development 과정에서 일관된 구조와 공통 언어 사용으로 반복 작업을 최소화하면 고객 만족도와 생산성이 향상되며, 개발 및 유지관리 비용이 절감되어 매출과 수익의 증가를 기대할 수 있습니다.

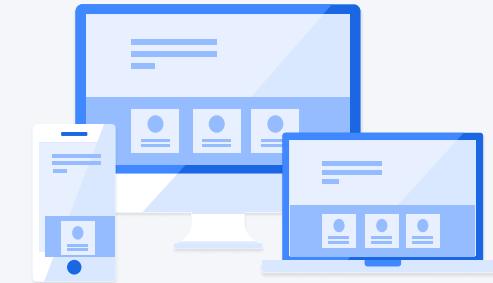
Research



UI Design & UX Modeling



Screen Development



1

명확한 가이드로
기획/디자인
효율 개선

2

컴포넌트를
라이브러리화 하여
개발효율 개선

3

일관된 UX
제공으로
사용자경험 개선

4

사후관리 및
기술지원으로
운영, 유지보수
쉬움

01

Research

기획 공통 가이드

스토리보드 작성에 자주 사용되는 모바일 화면 양식과 컴포넌트가 제공되어 작업자 별로 일관성 있는 산출물 작성이 가능합니다.

회원가입, 로그인 등의 공통 프로세스
샘플이 제공되어 업무 진행 시 활용할
수 있습니다.

This wireframe diagram illustrates the design of a mobile application interface, organized into several sections:

- Typography**: Describes the font used throughout the app.
- App Bar**: Shows the main page navigation bar with icons for message, home, and menu, and a sub-page navigation bar with back, forward, and search functions.
- Buttons**: Displays various button styles including Primary, Secondary, Optional, Disabled, Fixed, and Disabled.
- Content Box**: Features a two-column layout where the left column shows icons for mobile phone and ID card, and the right column displays text for mobile authentication and user registration.
- Control**: A section for input controls, showing a grid of checkboxes and radio buttons.
- Input**: A section for text input fields, including fields for name and password, with placeholder text and validation messages.
- List**: A section for lists, showing a hierarchical structure with checkmarks and arrows.
- Stepper**: A section for numerical input, showing a slider with step markers and a dropdown menu.

 기획 공통 가이드와 컴포넌트를 활용할 수 있습니다.

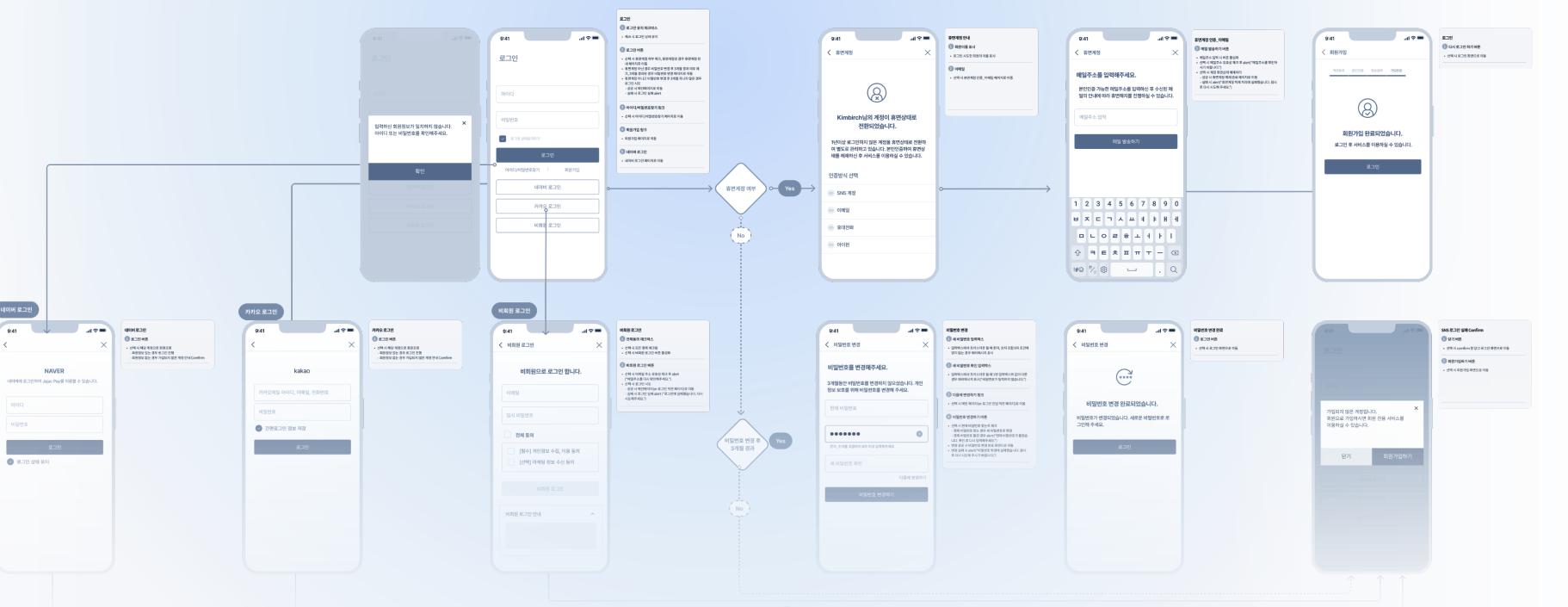
01

Research

Story Board + Flow Chart

스토리보드와 워크플로를 통합하여 전체 비즈니스 흐름을 한눈에 확인할 수 있습니다.

사용자 흐름을 파악하고 UI 설계를 점검해 볼 수 있습니다.



스토리보드와 워크플로를 통합한 산출물을 제공합니다.

02

Design & Modeling

UI Kit

디자인 목업을 위한 공통 UI 기준 정의서를 제공하여 디자인 작업의 업무 로드를 줄여줍니다.

가이드에 따라 작업 시 메인 컬러나 콘셉트가 변경되는 경우 바로 작업 결과에 적용해 볼 수 있습니다.

 디자인 정책과 스타일 가이드를 제공합니다.

 Jax Scripts for a neat and clever UI

Colors

23.12.13 Updated 

Primary colors

Jax-Gray

50	100	200	300	400	500	600	700	800	900
#F9FAFB	#EFF1F3	#E4E7EC	#CED2DA	#97A1AF	#637083	#41E62	#344051	#202B37	#141C24

Basic

white	Black
#FFFFFF	#000000

Status

success	warning	danger
#22C682	#EBAE67	#D96068

Jax-blue

50	100	200	300	400	500	600	700	800	900
#F1F6FF	#E8F0FF	#DAE8FF	#BDD6FF	#94BCFF	#408FFF	#1A66E3	#0E4DB5	#033282	#00104C

Jax-green

50	100	200	300	400	500	600	700	800	900
#F0FFF9	#D9FFEC	#ABFBD9	#67F088	#3BE59F	#22C682	#08A565	#318161	#24644A	#0E452E

Jax-turquoise

50	100	200	300	400	500	600	700	800	900
#EAFCFE	#CFF5F9	#A9EBF1	#6FDDE8	#0FC6D8	#0CB0CO	#0E95A3	#057C88	#015B64	#003E44

Jax-steelblue

50	100	200	300	400	500	600	700	800	900
#FBFBFF	#E9F1FF	#CEDF4	#B7CDF0	#84ABE5	#5B87CA	#4672B4	#2F498A	#172D64	#0B1C48

Jax-royalblue

50	100	200	300	400	500	600	700	800	900
#F5F5FF	#E2E0FF	#C9C5FF	#B3ACFE	#978EFF	#7760F2	#5E53DA	#4037A7	#271F85	#150F55

Jax-Red

50	100	200	300	400	500	600	700	800	900
#FFF5F5	#FCC0CF	#FB909D	#FA8585	#F76E6E	#F73B3B	#F200D0	#C20A0A	#910808	#610505

Type Scale

H1_Regular (28 / 140%)

A birch tree with many uses
쓰임새 많은 자작나무
12345

H1_Medium (28 / 10%)

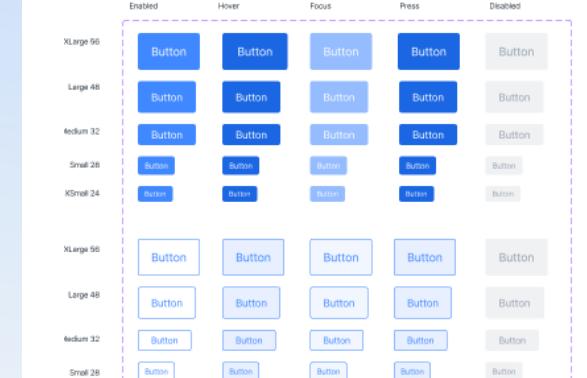
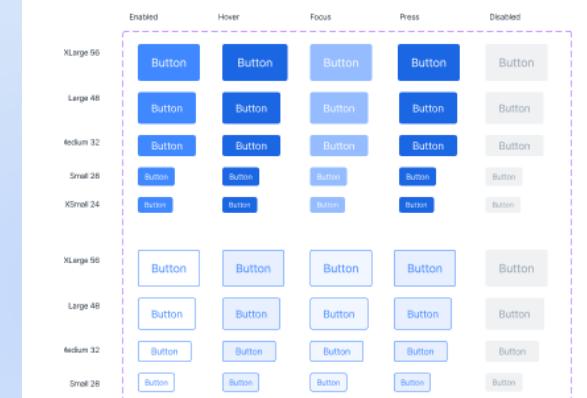
A birch tree with many uses
쓰임새 많은 자작나무
12345

H2_Regular (28 / 140%)

A birch tree with many uses
쓰임새 많은 자작나무
12345

H2_Medium (28 / 140%)

A birch tree with many uses
쓰임새 많은 자작나무
12345



Default

항목명	항목명
데이터	데이터
데이터	데이터
데이터	데이터

Multiple header

구분	항목명	항목명
	항목명	데이터

이름

플레이스홀더

비밀번호

••••••••••

휴대전화

통신사 ▾

인증

레이블

보안문자

48457

Default

Chip Text	Chip Text X

Removable

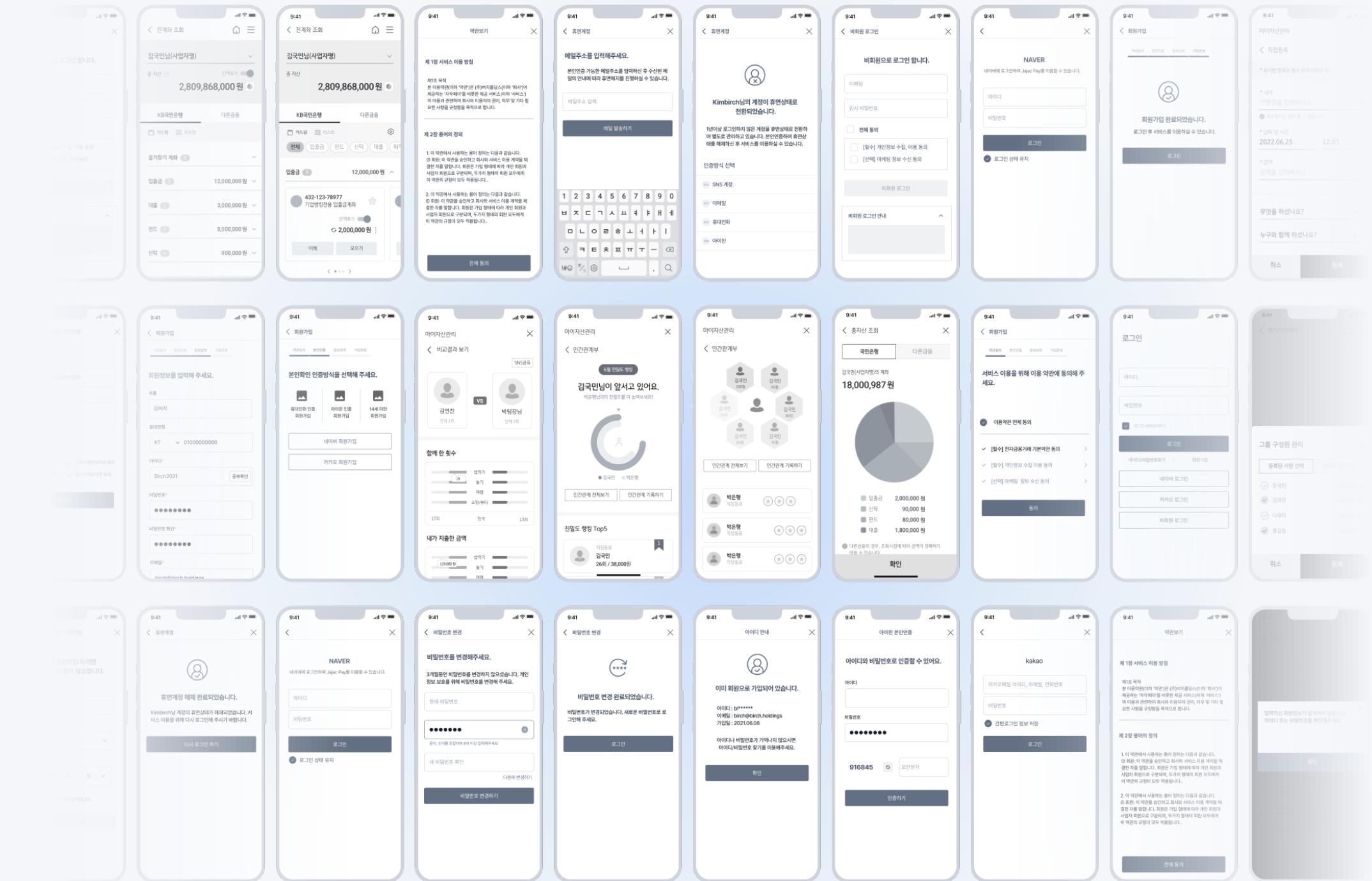
02

Design & Modeling

Style Guide

자주 사용되는 컴포넌트가 제공되어
빠르게 작업할 수 있습니다.

로그인, 회원가입 등 공통적으로 사용되는 UI는 jajac에서 제공되는 샘플 화면을 활용해서 쉽고 빠르게 만들 수 있습니다.



 디바이스별 목업 및 프로토타입을 쉽게 제작할 수 있습니다.

03

Screen Development

퍼블리싱 Index

작업 진행률과 진척 상태, 검수 여부를 확인할 수 있고 페이지별 변경 이력과 이슈를 체크할 수 있는 협업용 인덱스 페이지를 제공합니다.

프로젝트 완료 후에는 산출물로 활용되어 유지보수 시 참고할 수 있습니다.

The image displays several screenshots of mobile and web interfaces, illustrating the implementation of a publishing index. The top right shows a 'My Asset Management' app with a progress bar at 100%, a summary table, and a 'Current Savings' section. The middle left shows the 'JejuAir DCU' app with a navigation history table and a detailed component view. The bottom right shows a 'Korean Air' app with flight status and passenger information. These examples demonstrate how the publishing index is used to provide real-time project status, navigation history, and detailed component or user information.

협업용 퍼블리싱 인덱스를 제공합니다.

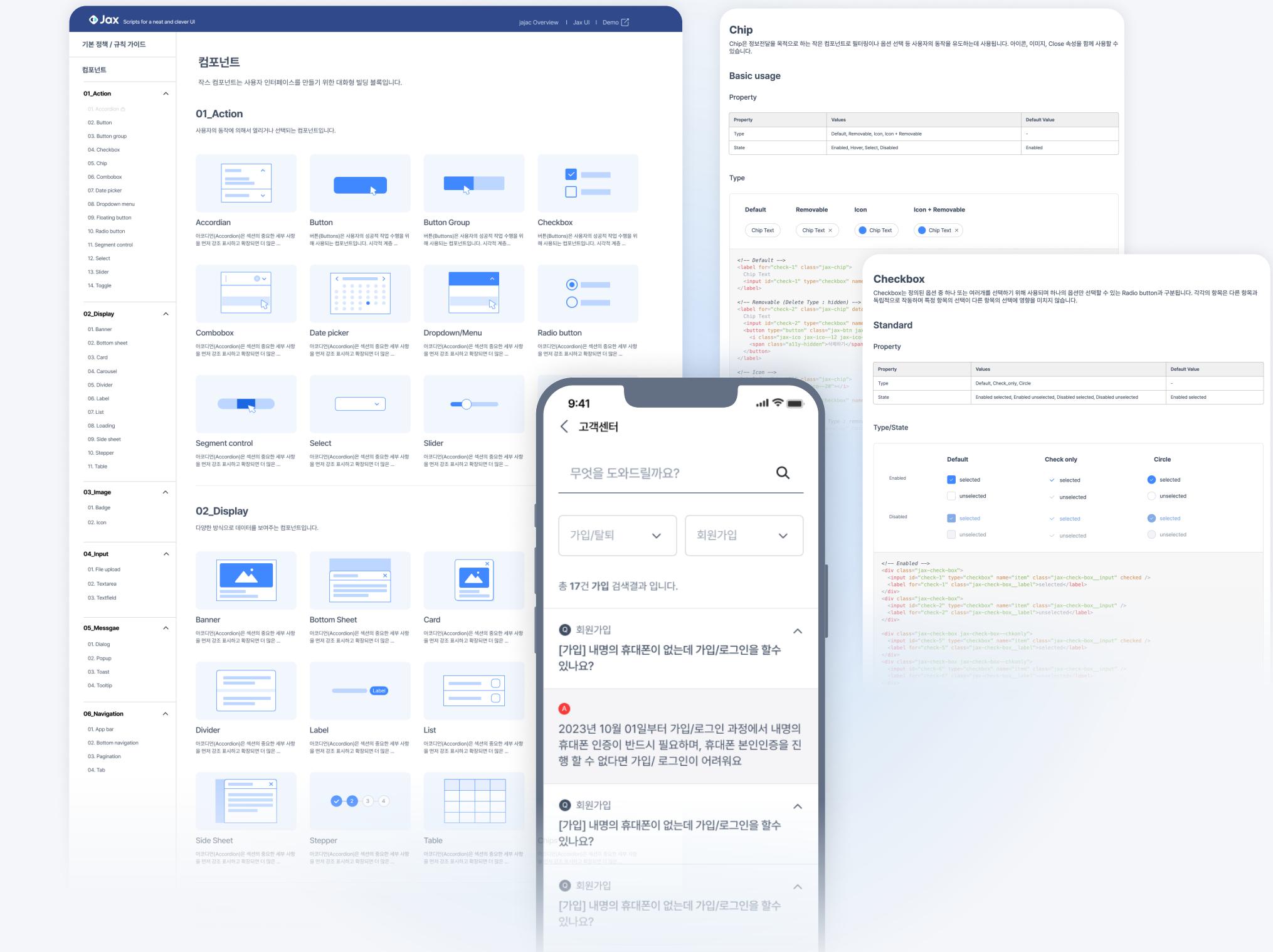
03

Screen Development

UI 프레임워크 nuts

스크립트 기반의 UI 프레임워크
 'nuts'에서 제공하는 다양한 컴포넌트
 를 활용하여 서비스를 개발하면 시간
 과 노력을 줄일 수 있습니다.

다양한 컴포넌트와 활용 가능한 데모
 템플릿을 제공합니다.



The image shows a screenshot of the Jax UI documentation and a mobile application interface. The documentation page is titled 'Jax UI' and includes sections for 'Basic usage' and 'Property' for various components like 'Accordion', 'Button', 'Checkbox', 'Combobox', 'Date picker', 'Dropdown menu', 'Floating button', 'Radio button', 'Segment control', 'Slider', 'Toggle', 'Display', 'Image', 'Input', 'Message', and 'Navigation'. It also features a 'Chip' component with code snippets and a mobile application interface showing a search bar, dropdown menus for '가입/탈퇴' and '회원가입', and a list of 17 search results. A detailed view of the 'Checkbox' component is shown on the right, including its basic usage, property table, and code examples for different states (Default, Removable, Icon, Icon + Removable).

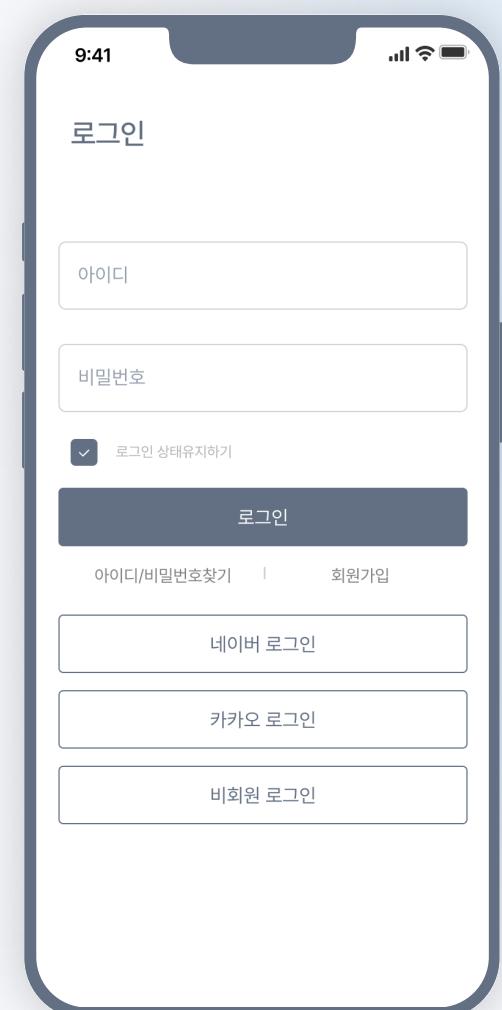
 nuts에서 제공하는 다양한 컴포넌트를 활용할 수 있습니다.

03

Screen Development

작업시간과 노력을 줄여주는 UI 프레임워크

'넛츠(nuts)'는 jajac 시스템에서 제공하는 스크립트 기반의 UI 프레임워크입니다. Vue, React, jQuery 등 다양한 자바스크립트 라이브러리와 호환이 가능하고, 기본이 되는 HTML 코드를 제공함으로써 소스의 통일성을 높일 수 있습니다. 다양한 컴포넌트를 제공하며 컴포넌트 단위로 조합이 가능하여 시간과 노력을 줄일 수 있습니다.



'넛츠'는 디자인시스템 '자작'에서 제공하는 스크립트 기반 UI Framework으로 Vue, React, jQuery 등 다양한 자바스크립트 라이브러리와 호환이 가능합니다.

UI for Vanilla JS

```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="utf-8" />
<!-- Meta Description -->
<meta name="description" content="버지 UI 프레임워크 'Jax'" />
<meta name="keywords" content="birch, jajac, Jax, 자작, 웹스, 디자인 시스템, UI 프레임워크" />
<meta name="author" content="버지홀딩스(Birch.)" />
<meta property="og:url" content="http://birch.holdings/" />
<meta property="og:title" content="Jax" />
<meta property="og:image" content="og-jax.png" />
<meta property="og:description" content="버지 UI 프레임워크 'Jax'" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width,initial-scale=1.0" />
<link rel="icon" href="favicon.ico" />
<link href="/jax/assets/style.min.css" rel="stylesheet" />
<link href="/index.css" rel="stylesheet" />
<ttitle>Jax Demo</ttitle>
</head>
<body>
<div id="app">
<div class="demo-container">
<div class="jax-app jax-app--login">
<h1 class="jax-app__title">jajac Pay</h1>
<main class="jax-contents">
<h2 class="jax-contents__title">로그인</h2>
<div class="form-control-box">
<div class="jax-form">
<div class="jax-form__group">
<input
  id="name-input-id"
  type="text"
  name="userId"
  class="jax-form__input jax-input"
  placeholder="아이디"
  title="아이디"
  data-jax-textfield
/>
<button type="button" class="jax-btn jax-btn--clear" data-jax-action-clear><span class="ally-hidden">지우기</span></button>
</div>
<div class="jax-form">
<div class="jax-form__group">
<input
  id="name-input-pw"
  type="password"
  name="userPw"
  class="jax-form__input jax-input"
  placeholder="패스워드"
  title="암호"
  data-jax-textfield
/>
<button type="button" class="jax-btn jax-btn--clear" data-jax-action-clear><span class="ally-hidden">지우기</span></button>
</div>
</div>
<div class="checkboxes">
<div class="checkbox">
<input checked="" type="checkbox" value="keepLoginFlag" />
<label for="check-keep" class="jax-check__label">로그인 상태유지</label>
</div>
</div>
<div class="form-control-box">
<div class="jax-form">
<div class="jax-form__group">
<input
  id="name-input-pw"
  type="password"
  name="userPw"
  class="jax-form__input jax-input"
  placeholder="패스워드"
  title="암호"
  data-jax-textfield
/>
<button type="button" class="jax-btn jax-btn--clear" data-jax-action-clear><span class="ally-hidden">지우기</span></button>
</div>
</div>
</div>
</div>
</div>
```

UI for React

```
import React from 'react';
const Login = () => {
  const [form, setForm] = React.useState({
    userId: '',
    userPw: '',
    keepLoginFlag: true,
  });
  const clickInputClearButton = () => {
    setForm({ ...form, userId: '' });
  };
  const changeInputValue = (e) => {
    setForm({ ...form, [e.target.name]: e.target.value });
  };
  const changeInputFlag = (e) => {
    setForm({ ...form, [e.target.name]: e.target.checked });
  };
  return (
    <div className="jax-app">
      <h1 className="jax-app__title">jajac Pay</h1>
      <main className="jax-contents">
        <h2 className="jax-contents__title">로그인</h2>
        <div className="form-control-box">
          <div className="jax-form">
            <div className="jax-form__group">
              <input
                id="name-input-id"
                type="text"
                name="userId"
                class="jax-form__input jax-input"
                placeholder="아이디"
                title="아이디"
                data-jax-textfield
              />
            </div>
            <div className="jax-form">
              <div className="jax-form__group">
                <input
                  id="name-input-pw"
                  type="password"
                  name="userPw"
                  class="jax-form__input jax-input"
                  placeholder="패스워드"
                  title="암호"
                  data-jax-textfield
                />
              </div>
            </div>
            <div className="checkboxes">
              <div className="checkbox">
                <input checked="" type="checkbox" value="keepLoginFlag" />
                <label for="check-keep" class="jax-check__label">로그인 상태유지</label>
              </div>
            </div>
          </div>
        </div>
      </main>
    </div>
  );
}
```

UI for Vue

```
<template>
<div class="jax-app jax-app--login">
  <h1 class="jax-app__title">jajac Pay</h1>
</div>
<main class="jax-contents">
  <div class="jax-contents__title">로그인</div>
  <div class="form-control-box form-control-box--checkbox">
    <div class="jax-form">
      <div class="jax-form__group">
        <input
          id="name-input-id"
          type="text"
          name="userId"
          class="jax-form__input jax-input"
          placeholder="아이디"
          title="아이디"
          data-jax-textfield
        />
      </div>
      <button type="button" class="jax-btn jax-btn--clear" data-jax-action-clear><span class="ally-hidden">지우기</span></button>
    </div>
    <div class="checkboxes">
      <div class="checkbox">
        <input checked="" type="checkbox" value="keepLoginFlag" />
        <label for="check-keep" class="jax-check__label">로그인 상태유지</label>
      </div>
    </div>
  </div>
</main>
</div>
```



Index 화면에서 프레임워크 별 코드를 확인할 수 있습니다.

04

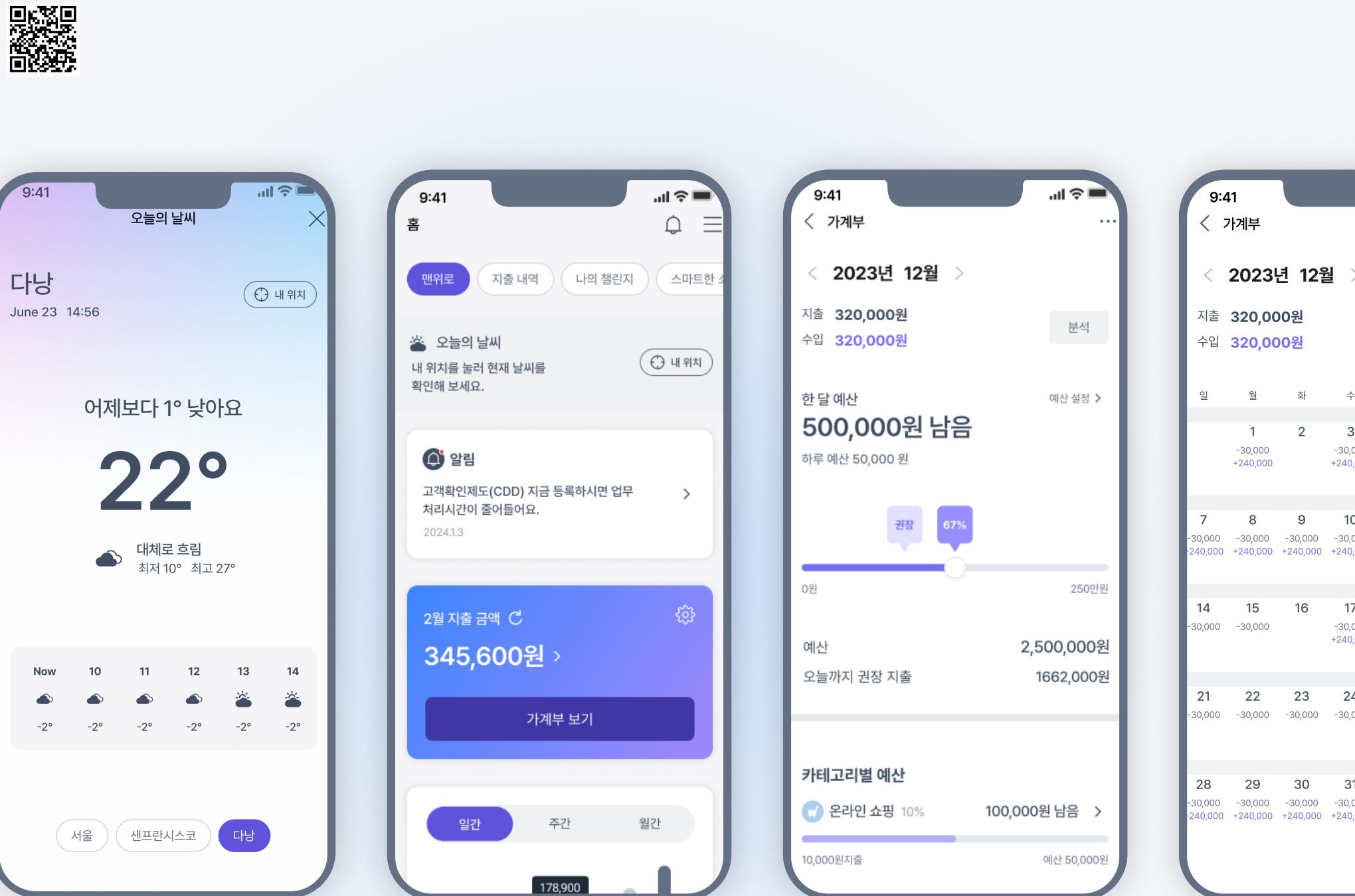
Case Study

jajac 금융 템플릿

금융 템플릿 홈 화면은 다양한 Card 컴포넌트를 사용하여 콘텐츠를 효과적으로 보여줍니다.

Slider를 활용해서 예산을 설정하고, 카테고리별 예산 사용 현황을 Progress bar로 표시 했습니다.

지역별 날씨는 Carousel을 사용하여 화면을 Swipe 할 때마다 지역이 변경되도록 했습니다.



!
지출 홈 화면과 가계부, 소비내역, 지역별 날씨 등의 템플릿을 제공합니다.

04

Case Study

jajac 스토리보드 템플릿

스토리보드 템플릿은 Card 컴포넌트로 현황을 한 눈에 볼 수 있도록 정리했습니다.

List 컴포넌트를 이용하여 좌측메뉴를 구성하고 Tooltip을 사용하여 각 항목에 대한 상세 설명을 제공합니다.



Card와 List를 다양하게 활용한 스토리보드 템플릿을 제공합니다.

Thanks

(주)버치홀딩스

TEL : +82-2-6092-0608 / FAX : 0504-044-0979 / E-Mail : contact@birch.holdings