

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: “Фронт-энд разработка”

**Отчет
Лабораторная работа №1**

Выполнил: Митурский Богдан Антонович

Группа: K33392

Санкт-Петербург

2023 г.

Цель:

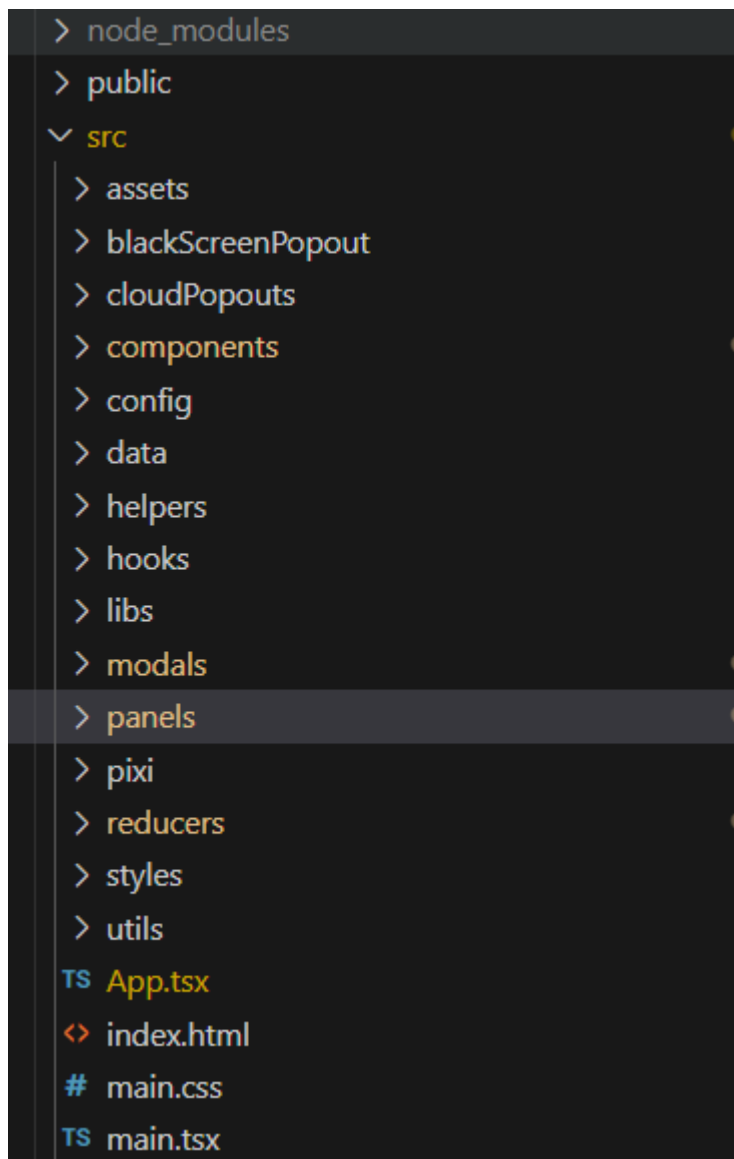
Реализовать верстку главной страницы для игры на площадке мини-приложений ВКонтакте



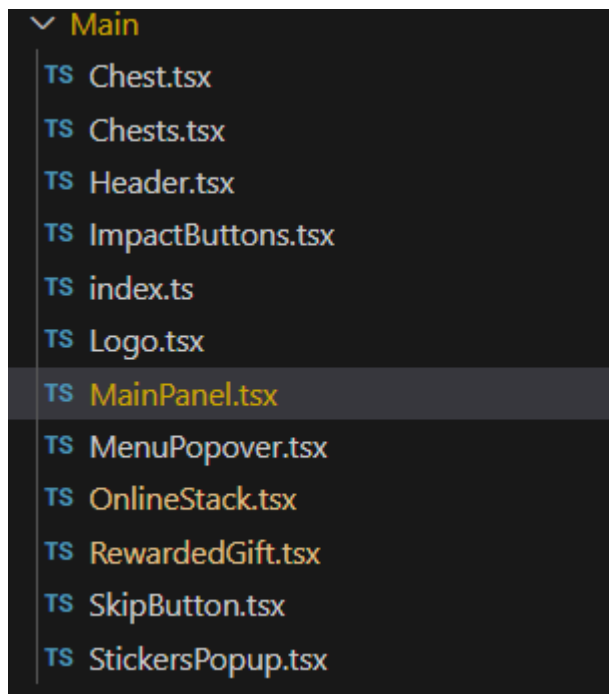
Программное обеспечение: TypeScript, React

Ход работы:

1. Развернем стандартное React приложение с помощью vite.
2. Продумаем структуру будущего приложения:



3. С помощью @emotion/styled и React-а создадим компоненты текущей страницы:



4. Напишем корневой файл страницы (MainPanel):

```
import React from "react";
import { Panel } from "@vkontakte/vkui";
import styled from "@emotion/styled";

import { images } from "../../data/url";
import Balance from "../../components/Balance";

import Header from "../Header";
import Logo from "../Logo";
import ImpactButtons from "../ImpactButtons";
import Chests from "../Chests";
import Snackbar from "@components/Snackbar";
import { pixiImagesLoaderStart } from "@hooks/pixiImagesLoader";

import OnlineStack from "../OnlineStack";
import StickersPopup from "../StickersPopup";
import bridge from "@vkontakte/vk-bridge";
import RewardedGift from "../RewardedGift";

// Кастомизируем панель из vkui
const Root = styled(Panel)`
  height: 100%;

  & > .vkuiPanel__in {
    height: 100%;
    background: url(${images.background.main}) center;
    background-size: cover;
```

```

    }
  `;

const Container = styled.div`
  height: 100%;
  display: flex;
  flex-direction: column;
  justify-content: space-between;
`;

const BottomGroup = styled.div`
  display: flex;
  flex-direction: column;
  align-items: center;
  gap: 16px;
`;

const HeaderRoot = styled.div``;

const DownHeaderContent = styled.div`
  display: flex;
  padding: 0 16px;
  padding-top: 16px;
  justify-content: space-between;
`;

type MainPanelProps = React.ComponentProps<typeof Panel>;

const MainPanel: React.FC<MainPanelProps> = () => {
  return (
    <Root>
      <Balance />
      <Container>
        <HeaderRoot>
          <Header />
          <DownHeaderContent>
            <OnlineStack />
          </DownHeaderContent>
        </HeaderRoot>
        <BottomGroup>
          <Logo />
          <ImpactButtons />
        </BottomGroup>
        <Chests />
      </Container>
      <Snackbar />
    </Root>
  );
};

```

```
};  
  
export default MainPanel;
```

5. Вынесем изображение логотипа в отдельный файл:

```
import React from "react";  
import styled from "@emotion/styled";  
  
import logo from "@img/logo.png";  
  
const LogoContainer = styled.div`  
  display: flex;  
  justify-content: center;  
  
  user-select: none;  
`;  
  
const LogoImg = styled.img`  
  width: 214px;  
`;  
  
const Logo = () => {  
  return (  
    <LogoContainer>  
      <LogoImg src={logo} alt="Логотип Sheep Royale" />  
    </LogoContainer>  
  );  
};  
  
export default React.memo(Logo);
```

6. Создадим на основе библиотеки vkui кастомизированный под свои нужды счетчик количества игроков:

```
import React, { FC, useEffect, useState, HTMLAttributes } from "react";  
import styled from "@emotion/styled";  
import { UsersStack } from "@vkontakte/vkui";  
  
import BorderedText from "../../components/BorderedText";  
import declination from "@/utils/declination";  
import getOnline from "@/utils/getOnline";  
import clamp from "@/utils/clamp";  
import getUserInfo from "@/utils/getUserInfo";  
import { useSelector } from "react-redux";  
import { RootState } from "@store";  
  
const Container = styled.div`  
  transition: 0.12s;
```

```

    height: 61px;
    display: flex;
    align-items: center;
    flex-direction: column;
  `;

const Label = styled(BorderedText)({
  color: "#ffffff",
  margin: 0,
});

type IOnlineStack = HTMLAttributes<HTMLDivElement>;

const OnlineStack: FC<IOnlineStack> = ({ ...props }) => {
  const [[count, users], setCount] = useState<[number, string[]]>([0, []]);
  const { _id } = useSelector((state: RootState) => state.User);

  const title =
    count >= 999
      ? "999+ ИГРОКОВ В СЕТИ"
      : `${count} ${declination(count, "ИГРОК", "ИГРОКА", "ИГРОКОВ")} В СЕТИ`;

  const styleOnline = {
    opacity: count ? "1" : "0",
  };

  return (
    <Container style={styleOnline} {...props}>
      <UsersStack visibleCount={3} layout="vertical" size="m" photos={users}
    />
    <Label key={title} component="span">
      {title}
    </Label>
  </Container>
);
};

export default React.memo(OnlineStack);

```

7. Сверстаем блок с сундуками:

```

import React, { useRef, useState } from "react";
import styled from "@emotion/styled";

import { useDispatch, useSelector } from "react-redux";
import { RootDispatch, RootState } from "@store";
import { push, useLocation, useMeta, replace } from "@itznevikat/router";
import { request } from "@utils/request";

```

```

import { IChest, setChests } from "@reducers/user.reducer";
import Chest from "../Chest";
import { soundEffects } from "@data/sounds/interface";
import bridge, { EAdsFormats } from "@vkontakte/vk-bridge";

import chestsBackgroundImg from "../ChestOpening/img/background.png";
import chestsFilterImg from "../ChestOpening/img/filter.png";
import { tutorialReplace } from "@modals/modalTutorial/blockedNavigation";
import { DataTutorialIds } from "@modals/modalTutorial/stepsConfig";

const Root = styled.div`
  display: flex;
  flex-direction: row;
  width: 100%;
  gap: 8px;
  margin-bottom: 12px;
  justify-content: center;
  max-height: 124px;
  user-select: none;
  position: relative;
  padding: 0 8px;
`;

const Chests = () => {
  return (
    <Root>
      {chests.map((chest, i) => (
        <Chest
          data-tutorial-step={!i ? DataTutorialIds.Chest : -1}
          loading={loadingIndex === i}
          available={loadingIndex === null}
          data={chest}
          key={i}
          onClick={() => handleChestClick(chest, i)}
          imageRef={(ref) => (imageRefs.current[i] = ref)}
        />
      ))}
    </Root>
  );
};

export default React.memo(Chests);

```

8. И также сверстаем отдельно и сам компонент сундука:

```

import React, { useState, useEffect, useMemo, useRef } from "react";
import BorderedText from "@components/BorderedText";
import { IChest } from "@reducers/user.reducer";
import { css, keyframes } from "@emotion/react";

```



```

import styled from "@emotion/styled";
import { ChestsData } from "../../../../../backend/src/data/chests";

import glow from "@img/chests/glow.png";
import SkipButton from "./SkipButton";
import formatWordByNumber from "@/utils/formatWordByNumber";
import { ChestsImages } from "@data/chests";
import { Spinner as VKUISpinner, Tappable } from "@vkontakte/vkui";
import anime from "animejs";

type IStatus = "idle" | "ready" | "opening";

const styles: Record<
  IStatus | "empty",
  {
    border: string;
    transparent?: boolean;
    background: string;
    shadow: string;
    innerShadow?: string;
  }
> = {
  empty: {
    border: "rgba(0, 0, 0, 0.05)",
    transparent: true,
    background: "rgba(0, 0, 0, 0.01)",
    shadow: "0px 3px 0px rgba(0, 0, 0, 0.05)",
    innerShadow:
      "inset 0px 8px 24px rgba(0, 0, 0, 0.09), inset 0px 0px 8px rgba(0, 0, 0, 0.05)",
  },
  idle: {
    border: "#85AEFF",
    transparent: true,
    background: "rgba(92, 177, 255, 0.25)",
    shadow: "0px 3px 0px #667A14",
    innerShadow:
      "inset 0px 8px 24px rgba(133, 145, 255, 0.5), inset 0px 0px 8px #85e9ff",
  },
  ready: {
    border: "#31CEFF",
    background: "#1196DE",
    shadow: "0px 3px 0px #375978",
  },
  opening: {
    border: "#29D87A",
    background:

```

```

        "linear-gradient(205.72deg, rgba(75, 255, 169, 0.195) 6.12%, rgba(43,
122, 108, 0.5) 92.82%)",
        shadow: "0px 3px 0px #2D7888",
    },
};

const BorderedTextSmall = styled(BorderedText)`
    font-size: 12px;
    line-height: 14px;
    white-space: nowrap;
`;

const Wrapper = styled(Tappable, {
    shouldForwardProp: (p) =>
        !["loading", "skipButton", "status"].some((e) => e === p),
})<{
    status?: IStatus;
    skipButton?: boolean;
    loading: boolean;
    clickable: boolean;
}>`
    position: relative;
    aspect-ratio: 82 / 124;
    flex: 1;
    max-width: 86px;
    border-radius: 14px;
    font-family: "Russo One";
    background: ${(props) =>
        styles[props.status || "empty"].transparent
        ? "rgba(0, 0, 0, 0.07)"
        : styles[props.status || "empty"].background};
    border: 2px solid ${(props) => styles[props.status || "empty"].border};
    padding: 3px;
    text-align: center;
    font-size: ${(props) =>
        props.status === "idle" ? "16px" : props.status ? "14px" : "12px"};
    color: #ffffff;
    box-shadow: ${(props) => styles[props.status || "empty"].shadow};
    cursor: ${(props) => (props.clickable ? "pointer" : "unset")};
    ${(props) => props.skipButton && "padding-bottom: 24px;"}
    transition: transform 0.12s;

    &.clicked {
        transform: scale(0.93);
    }
`;

const Content = styled.div<{ status?: IStatus }>`

```

```

width: 100%;
height: 100%;
border-radius: 10px;
display: flex;
justify-content: center;
align-items: center;
flex-direction: column;
background: ${props =>
  styles[props.status || "empty"].transparent &&
  styles[props.status || "empty"].background};
padding: 8px;

box-shadow: ${props => styles[props.status || "empty"].innerShadow};

${props =>
  !props.status &&
  css`
    & > div {
      filter: drop-shadow(0px 1px 0px rgba(0, 0, 0, 0.24))
        drop-shadow(0px -1px 0px rgba(0, 0, 0, 0.24))
        drop-shadow(1px 0px 0px rgba(0, 0, 0, 0.24))
        drop-shadow(-1px 0px 0px rgba(0, 0, 0, 0.24));
      white-space: pre-wrap;
      letter-spacing: 0.28px;
    }
  `}
`;

```

```

const Image = styled.img<{ status: IStatus }>`
  width: 95%;
  display: block;
  position: relative;
  z-index: 1;

  ${props =>
    props.status === "idle" &&
    css`
      margin-bottom: 8px;
    `}

  ${props =>
    props.status === "opening" &&
    css`
      margin-top: 8px;
    `}

  ${props =>
    props.status === "ready" &&

```

```

    css`
      margin-bottom: 8px;
    `
  `;

const ImageGlowWrapper = styled.div`
  position: relative;
  display: flex;
  align-items: center;
  justify-content: center;
  z-index: 0;
`;

const glowAnimations = keyframes`
  0% {
    transform: rotate(0deg);
  }

  100% {
    transform: rotate(360deg);
  }
`;

const Glow = styled.img`
  position: absolute;
  animation: ${glowAnimations} 60s infinite linear;
`;

const Spinner = styled(VKUISpinner)`
  color: #e5e6ff;
`;

const Chest = ({
  return (
    <Wrapper
      status={status}
      skipButton={skipButton}
      loading={loading}
      hasActive={clickable}
      hasHover={clickable}
      clickable={clickable}
      activeMode="clicked"
      className={loading ? "clicked" : ""}
      {...props}
    >
    <Content status={status}>
      {data === null && <div>{"Место\ндля\нсундука"}</div>}
    </Content>
  )
}

```

```

    {data && status === "idle" && (
      <>
        <Image ref={imageRef} src={chestImage} status={status} />
        <BorderedText>{time}</BorderedText>
      </>
    )}

    {data && status === "opening" && (
      <>
        <BorderedTextSmall ref={timeTitleRef}>{time}</BorderedTextSmall>
        <Image ref={imageRef} src={chestImage} status={status} />
        {skipButton && (
          <SkipButton
            available={available}
          >{`Ускорить - ${adWatchBonus}`}</SkipButton>
        )}
      </>
    )}

    {data && status === "ready" && !loading && (
      <>
        <ImageGlowWrapper>
          <Image ref={imageRef} src={chestImage} status={status} />
          <Glow src={glow} />
        </ImageGlowWrapper>
        <BorderedText>Открыть</BorderedText>
      </>
    )}

    {loading && <Spinner />}
  </Content>
</Wrapper>
);
};

export default Chest;

```

9. Сверстаем главные кнопки:

```

import React, { useState } from "react";
import styled from "@emotion/styled";

import { menuButtonSound } from "../../data/sounds/interface";
import ImpactButton from "../../components/ImpactButton";
import { push, useLocation, useParams, useMeta } from "@itznevikat/router";
import { useSelector, useStore } from "react-redux";
import { RootState } from "@store";
import UserAdd from "@components/svg/UserAdd";

```

```

import bridge, {
  BannerAdLayoutType,
  BannerAdLocation,
} from "@vkontakte/vk-bridge";
import useShowBannerAd from "@/hooks/useShowBannerAd";

const Root = styled.div`
  display: flex;
  justify-content: center;
  align-items: center;
  gap: 8px;
`;

type HomeScreenButtonType = {
  fontSize?: 24 | 20 | 16;
  background?: string;
  before?: boolean;
};

type Meta = {
  gameMode: string;
};

const HomeScreenButton = styled(ImpactButton)<HomeScreenButtonType>(
  ({ fontSize, before }) => ({
    "& .ImpactButton__title": {
      fontSize: fontSize,
      paddingBottom: 0,
    },
    "& .ImpactButton__title::before": {
      top: before ? undefined : 1,
    },
    "& .ImpactButton__content": {
      padding: "0px",
    },
  }),
  ({ background }) => ({
    "& .ImpactButton__content": {
      background: background,
    },
  }),
  () => ({
    "& .ImpactButton__container": {
      display: "flex",
      justifyContent: "space-between",
      alignItems: "center",
      gap: 8,
    },
  }),

```

```

    })
  );

const ImpactButtons = () => {

  return (
    <>
      <Root>
        <HomeScreenButton
          onClick={handlePlayButtonOneOnOneBattle}
          disabled={!cloudsPopout && !meta?.gameMode}
          data-tutorial-step={1}
          mode="primary"
          fontSize={24}
          before
        >
          {currentProgress === null ? "Играть" : `Загрузка
${currentProgress}%`}
        </HomeScreenButton>
        <HomeScreenButton
          onClick={handlePlayButtonCreateLobby}
          disabled={!cloudsPopout && meta?.gameMode === "2vs2"}
          mode="secondary"
          fontSize={18}
          subtitle="Лобби, без призов"
        >
          {currentProgress === null
            ? "Создать"
            : `Загрузка ${currentProgress}%`}
        </HomeScreenButton>
      </Root>
    </>
  );
};

export default React.memo(ImpactButtons);

```

10. И добавим кнопки вверху экрана:

```

import React from "react";
import styled from "@emotion/styled";

import {
  Icon24Users30Outline,
  Icon24MessagesOutline,
  Icon24MenuOutline,
} from "@vkontakte/icons";

import { push, useLocation, useParams, back } from "@itznevikat/router";

```

```

import urls, { icons } from "../../data/url";

import BorderedText from "../../components/BorderedText";
import ButtonWithHighlight from "../../components/ButtonWithHighlight";
import MenuPopover from "../MenuPopover";

import { soundEffects } from "../../data/sounds/interface";

const Root = styled.div`
  margin: 56px 16px 0px 16px;
  display: flex;
  justify-content: space-between;
  align-items: flex-start;
  gap: 4px;
`;

export const BonusButton = styled(ButtonWithHighlight)`
  height: 44px;
  max-width: 256px;
  padding: 8px;

  justify-content: flex-start;
  flex-grow: 1;
  gap: 8px;

  font-family: Russo One !important;
  font-size: 16px;
  letter-spacing: 0.28px;

  background: linear-gradient(180deg, #efd02a 0%, #efb82a 100%);
  box-shadow: 0px 2px 0px #974d07, 0px 4px 0px rgba(0, 0, 0, 0.32),
    inset 0px -0.5px 0px rgba(255, 255, 255, 0.3),
    inset 0px 0.5px 0px rgba(255, 255, 255, 0.5);

  user-select: none;
`;

export const BonusIcon = styled.img`
  width: 20px;
  height: 20.64px;
`;

const ButtonGroup = styled.span`
  display: flex;
  gap: 4px;
`;

```



```

const Component = styled(ButtonWithHighlight)`
  width: 42px;
  height: 44px;

  text-decoration: none;
  color: white;
  outline: none;

  & svg {
    filter: drop-shadow(0px 0.5px 0px #d9fdff);
  }
`;

type IconButtonProps = React.ComponentProps<"a">;
const IconButton: React.FC<IconButtonProps> = ({ children, ...props }) => {
  return (
    <Component component="a" target="_blank" rel="noreferrer" {...props}>
      {children}
    </Component>
  );
};

const Header = () => {
  return (
    <Root>
      <BonusButton onClick={handleBonusButtonClick}>
        <BonusIcon src={icons.gift} />
        <BorderedText shadowColor="#302508" borderColor="#302508">
          Бонусы
        </BorderedText>
      </BonusButton>
      <ButtonGroup>
        <IconButton href={urls.group} onClick={handleIconButtonClick}>
          <Icon24Users30Outline />
        </IconButton>
        <IconButton href={urls.chat} onClick={handleIconButtonClick}>
          <Icon24MessagesOutline />
        </IconButton>
        <ButtonWithHighlight
          forwardedRef={menuButtonRef}
          onClick={handleMenuButtonClick}
        >
          <Icon24MenuOutline />
        </ButtonWithHighlight>
      </ButtonGroup>
      <MenuPopover isOpen={!isMenuOpened} elementRef={menuButtonRef} />
    </Root>
  );
};

```

```
};  
  
export default React.memo(Header);
```

Вывод:

В ходе работы я заготовил структуру будущего проекта и реализовал верстку главного экрана согласно макету из задачи.