##Контекст проекта

##Структура

```
.gitignore
frontend/
  index.html
  package-lock.json
  package.json
  public/
    logo.svg
  src/
    api.js
    App.jsx
    components/
      AddAssetForm.jsx
      AssetsTable.jsx
      CoinInfo.jsx
      CoinInfoModal.jsx
      DarkSpotEffect.jsx
      layout/
        AppContent.jsx
        AppHeader.jsx
        AppLayout.jsx
        AppSider.jsx
        styles/
          AppContent.css
          AppHeader.css
          AppSider.css
      PortfolioChart.jsx
      styles/
        AssetsTable.css
    context/
      crypto-context.jsx
    data.js
    index.css
    main.jsx
    utils.js
  vite.config.js
package-lock.json
package.json
server.js
```

##Сниппеты файлов

frontend\index.html

```
<!doctype html>
<html lang="ru">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/logo.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Crypto Wallet</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

frontend\package.json

```
{
  "name": "react-crypto-app",
  "private": true,
  "version": "0.0.0",
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "lint": "eslint . --ext js,jsx --report-unused-disable-directives --max-warn
    "preview": "vite preview"
  },
  "dependencies": {
    "antd": "^5.24.6",
    "react": "^18.2.0",
    "react-chartjs-2": "^5.3.0",
    "react-dom": "^18.2.0"
  },
  "devDependencies": {
    "@types/react": "^18.2.43",
    "@types/react-dom": "^18.2.17",
    "@vitejs/plugin-react": "^4.2.1",
    "eslint": "^8.55.0",
    "eslint-plugin-react": "^7.33.2",
    "eslint-plugin-react-hooks": "^4.6.0",
    "eslint-plugin-react-refresh": "^0.4.5",
    "vite": "^5.0.8"
  }
}
```

frontend\src\api.js

```
import { cryptoAssets, cryptoData } from "./data";

export function fakeFetchCrypto() {
    return new Promise(resolve => {
        setTimeout(() => {
            resolve(cryptoData)
        }, 1)
    });
};

export function fetchAssets() {
    return new Promise(resolve => {
        setTimeout(() => {
            resolve(cryptoAssets)
        }, 1)
    });
};
```

frontend\src\App.jsx

```
import AppLayout from "./components/layout/AppLayout";
import { CryptoContextProvider } from "./context/crypto-context";

export default function App() {
  return (
    <CryptoContextProvider>
      <AppLayout />
    </CryptoContextProvider>
  );
}
```

frontend\src\components\AddAssetForm.jsx

```
import { Select, Space, Typography, Flex, Divider, Form, InputNumber, Button, Da
import { useState, useRef } from "react"
import { useCrypto } from "../context/crypto-context"
import CoinInfo from "./CoinInfo"

const validateMessages = {
    required: '${label} is required!',
    types: {
        number: '${label} is not valid number',
    },
    number: {
        range: '${label} must be between ${min} and ${max}',
    },
}

export default function AddAssetForm({ onClose }) {
    const [form] = Form.useForm()
    const {crypto, addAsset} = useCrypto()
    const [coin, setCoin] = useState(null)
    const [submitted, setSubmitted] = useState(false)
    const assetRef = useRef()

    if (submitted) {
        return (
            <Result
                status="success"
                title="New Asset Added"
                subTitle={`Added ${assetRef.current.amount} of ${coin.name} by p
                extra={[
                    <Button type="primary" key="console" onClick={onClose}>
                        Close
                    </Button>
                ]}
            />
        )
    }

    if (!coin) {
        return (
            <Select
                style={{ width: '100%' }}
                onSelect={(v) => setCoin(crypto.find((c) => c.id === v))}
                //value='Press "/" to search'
                placeholder='Выберите валюту'
                options={crypto.map(coin => ({
                    label: coin.name,
                    value: coin.id,
                    icon: coin.icon,
                }))}
                optionRender={option => (
                    <Space>
                        <img style={{width: 20}} src={option.data.icon} alt={opt
                        {option.data.label}
                    </Space>
                )}
            >
            </Select>
        )
    }

    function onFinish(values) {
        const newAsset = {
            id: coin.id,
            amount: values.amount,
            price: values.price,
            date: values.date?.$d ?? new Date(),
        }
        assetRef.current = newAsset
        setSubmitted(true)
        addAsset(newAsset)
    }

    function handleAmountChange(value) {
        const price = form.getFieldValue('price')
        form.setFieldsValue({
            total: +(value * price).toFixed(2),

        })
    }

    function handlePriceChange(value) {
        const amount = form.getFieldValue('amount')
        form.setFieldsValue({
            total: +(amount * value).toFixed(2),

        })
    }

    return (
        <Form
            form={form}
            name="basic"
            labelCol={{ span: 4 }}
            wrapperCol={{ span: 10 }}
            style={{ maxWidth: 600 }}
            initialValues={{
                price: +coin.price.toFixed(2),
            }}
            onFinish={onFinish}
            validateMessages={validateMessages}
        >
```

```
            <CoinInfo coin={coin} />
            <Divider />

            <Form.Item
                label="Количество"
                name="amount"
                rules={[{
                    required: true,
                    type: 'number',
                    min: 0,
                }]}
            >
                <InputNumber
                    placeholder="Введите количество"
                    style={{width: '100%'}}
                    onChange={handleAmountChange}
                />
            </Form.Item>

            <Form.Item
                label="Цена"
                name="price"
            >
                <InputNumber onChange={handlePriceChange} style={{width: '100%'}
            </Form.Item>

            <Form.Item
                label="Дата покупки"
                name="date"
            >
                <DatePicker placeholder="Выберите дату" style={{width: 230}}/>
            </Form.Item>

            <Form.Item
                label="Итог"
                name="total"
            >
                <InputNumber disabled style={{width: '100%'}} />
            </Form.Item>

            <Form.Item label={null}>
                <Button type="primary" htmlType="submit" style={{ backgroundColo
                    Добавить
                </Button>
            </Form.Item>
        </Form>
    )
}
```

frontend\src\components\AssetsTable.jsx

```
import { Table } from "antd";
import { useCrypto } from "../context/crypto-context";
import './styles/AssetsTable.css'

const columns = [
  {
    title: "Название",
    dataIndex: "name",
    showSorterTooltip: { target: "full-header" },
    sorter: (a, b) => a.name.length - b.name.length,
    sortDirections: ["descend"],
  },
  {
    title: "Цена, $",
    dataIndex: "price",
    defaultSortOrder: "descend",
    sorter: (a, b) => a.price - b.price,
  },
  {
    title: "Количество",
    dataIndex: "amount",
    sorter: (a, b) => a.amount - b.amount,
  },
];

export default function AssetsTable() {
  const { assets } = useCrypto();
  const data = assets.map((a) => ({
    key: a.id,
    name: a.name,
    price: a.price,
    amount: a.amount,
  }));
  return (
    <Table
      className="table"
      sticky={true}
      pagination={false}
      columns={columns}
      dataSource={data}
    />
  );
}
```

frontend\src\components\CoinInfo.jsx

```
import { Flex, Typography } from "antd"

export default function CoinInfo({ coin, withSymbol }) {
    return (
        <Flex align="center">
                <img
                    src={coin.icon}
                    alt={coin.name}
                    style={{width: 40, marginRight: 10}}
                />
                <Typography.Title level={2} style={{ margin: 0 }} >
                    {withSymbol && <span>({coin.symbol})</span>} {coin.name}
                </Typography.Title>
        </Flex>
    )
}
```

frontend\src\components\CoinInfoModal.jsx

```
import { Divider, Flex, Tag, Typography } from "antd"
import CoinInfo from "./CoinInfo"

export default function CoinInfoModal({ coin }) {
    return (
        <>
            <CoinInfo coin={coin} withSymbol />
            <Divider />
            <Typography.Paragraph>
                <Typography.Text strong>1 час: </Typography.Text>
                <Tag color={coin.priceChange1h > 0 ? 'green' : 'red' }>{coin.pri
                <Typography.Text strong>1 день: </Typography.Text>
                <Tag color={coin.priceChange1d > 0 ? 'green' : 'red' }>{coin.pri
                <Typography.Text strong>1 неделя: </Typography.Text>
                <Tag color={coin.priceChange1w > 0 ? 'green' : 'red' }>{coin.pri
            </Typography.Paragraph>
            <Typography.Paragraph>
                <Typography.Text strong>Цена: </Typography.Text>
                {coin.price.toFixed(2)}$
            </Typography.Paragraph>
            <Typography.Paragraph>
                <Typography.Text strong>Цена в BTC: </Typography.Text>
                {coin.priceBtc}
            </Typography.Paragraph>
            <Typography.Paragraph>
                <Typography.Text strong>Капитализация: </Typography.Text>
                {coin.marketCap}$
            </Typography.Paragraph>
            {coin.contractAddress && <Typography.Paragraph>
                <Typography.Text strong>Адрес контракта: </Typography.Text>
                {coin.contractAddress}
            </Typography.Paragraph>}
        </>
    )
}
```

frontend\src\components\DarkSpotEffect.jsx

```
import { useEffect, useRef, useState } from 'react';

const DarkSpotEffect = () => {
  const [position, setPosition] = useState({ x: -100, y: -100 }); // начальное п
  const size = 50; // размер пятна

  useEffect(() => {
    const handleMouseMove = (e) => {
      setPosition({ x: e.clientX, y: e.clientY });
    };

    window.addEventListener('mousemove', handleMouseMove);
    return () => window.removeEventListener('mousemove', handleMouseMove);
  }, []);

  return (
    <div
```

```
      style={{
        position: 'fixed',
        top: position.y - size / 2,
        left: position.x - size / 2,
        width: size,
        height: size,
        backgroundColor: 'rgba(19, 47, 73, 0.1',
        borderRadius: '50%',
        filter: 'blur(20px)',
        pointerEvents: 'none',
        zIndex: 9999,
        transition: 'top 0.02s, left 0.02s',
      }}
    />
  );
};

export default DarkSpotEffect;
```

frontend\src\components\layout\AppContent.jsx

```
import { Layout, Typography } from "antd";
import { useCrypto } from "../../context/crypto-context";
import PortfolioChart from "../PortfolioChart";
import AssetsTable from "../AssetsTable";
import './styles/AppContent.css'


export default function AppContent() {
  const { assets, crypto } = useCrypto();

  const cryptoPriceMap = crypto.reduce((acc, c) => {
    acc[c.id] = c.price;
    return acc;
  }, {});

  return (
    <Layout.Content className="content">
      <PortfolioChart style={{marginTop: 200}}/>
      <Typography.Title
        level={3}
        className="title"
        style={{ textAlign: "left", color: "#646464" }}
      >
        Общая цена портфеля: {""}
        {assets
          .map((asset) => asset.amount * cryptoPriceMap[asset.id])
          .reduce((acc, v) => (acc += v), 0)
          .toFixed(2)}
        $
      </Typography.Title>
      <AssetsTable />
    </Layout.Content>
  );
}
```

frontend\src\components\layout\AppHeader.jsx

```
import { Layout, Select, Space, Button, Modal, Drawer } from "antd";
import { useCrypto } from "../../context/crypto-context";
import { useEffect, useState } from "react";
import CoinInfoModal from "../CoinInfoModal";
import AddAssetForm from "../AddAssetForm";
import "./styles/AppHeader.css";

export default function AppHeader() {
  const [select, setSelect] = useState(false);
  const [coin, setCoin] = useState(null);
  const [modal, setModal] = useState(false);
  const [drawer, setDrawer] = useState(false);
  const { crypto } = useCrypto();

  useEffect(() => {
    const keypress = (event) => {
      if (event.key === "/") {
        setSelect((prev) => !prev);
      }
    };
    document.addEventListener("keypress", keypress);
    return () => document.removeEventListener("keypress", keypress);
  }, []);

  function handleSelect(value) {
    setCoin(crypto.find((c) => c.id === value));
    setModal(true);
  }

  return (
    <Layout.Header className="header">
      <Select
        style={{ width: "250px" }}
        open={select}
        onSelect={handleSelect}
        onClick={() => setSelect((prev) => !prev)}
        value='Нажмите "/" для поиска'
        options={crypto.map((coin) => ({
          label: coin.name,
          value: coin.id,
          icon: coin.icon,
        }))}
        optionRender={(option) => (
          <Space>
            <img
              style={{ width: 20 }}
              src={option.data.icon}
              alt={option.data.label}
            />
            {""}
            {option.data.label}
          </Space>
        )}
      ></Select>

      <a href="#">
        <div className="wrapper">
          <img className="logo" src="logo.svg" alt="logo" />
          <h1 className="title">CryptoWallet</h1>
        </div>
      </a>

      <Button
        type="primary"
        onClick={() => setDrawer(true)}
        style={{ backgroundColor: "#132F49" }}
      >
        Добавить транзакцию
      </Button>

      <Modal open={modal} onCancel={() => setModal(false)} footer={null}>
        <CoinInfoModal coin={coin} />
      </Modal>

      <Drawer
        style={{ backgroundColor: "#FAFAFA" }}
        width={600}
        title="Добавить транзакцию"
        onClose={() => setDrawer(false)}
        open={drawer}
        destroyOnClose
      >
        <AddAssetForm onClose={() => setDrawer(false)} />
      </Drawer>
    </Layout.Header>
  );
}
```

frontend\src\components\layout\AppLayout.jsx

```
import { useContext } from "react";
import CryptoContext from "../../context/crypto-context";
import { Layout, Spin } from "antd";
import AppHeader from "./AppHeader";
import AppSider from "./AppSider";
import AppContent from "./AppContent";
import DarkSpotEffect from "../DarkSpotEffect.jsx";

export default function AppLayout() {
  const layoutStyle = {
    maxHeight: '100vh',
    overflow: 'hidden',
  }

  const { loading } = useContext(CryptoContext);

  if (loading) {
    return <Spin fullscreen />;
  }

  return (
    <Layout style={layoutStyle}>
      <DarkSpotEffect />
      <AppHeader />
      <Layout>
```

```
        <AppSider />
        <AppContent />
      </Layout>
    </Layout>
  );
}
```

frontend\src\components\layout\AppSider.jsx

```
import { Layout, Card, Statistic, List, Typography, Tag } from 'antd';
import { ArrowDownOutlined, ArrowUpOutlined } from '@ant-design/icons';
import { capitalize } from '../../utils';
import { useContext } from 'react';
import CryptoContext from '../../context/crypto-context';
import './styles/AppSider.css'


export default function AppSider() {
    const {assets} = useContext(CryptoContext)

    return (
        <Layout.Sider width="25%" className='sider'>
            {assets.map(asset => (
                <Card key={asset.id} style={{ marginBottom: '1rem', backgroundCo
                <Statistic
                    title={capitalize(asset.id)}
                    value={asset.totalAmount}
                    precision={2}
                    valueStyle={{ color: asset.grow ? '#3f8600' : '#cf1322' }}
                    prefix={asset.grow ? <ArrowUpOutlined /> : <ArrowDownOutline
                    suffix="$"
                />
                <List
                    size='small'
                    dataSource={[
                        { title: 'Прибыль:', value: asset.totalProfit, withTag:
                        { title: 'Количество:', value: asset.amount, isPlain: tr
                    ]}
                    renderItem={(item) => (
                    <List.Item>
                        <span>{item.title}</span>
                        <span>
                            {item.withTag && (
                                <Tag color={asset.grow ? 'green' : 'red'}>{asset
                            )}
                            {item.isPlain && item.value}
                            {!item.isPlain && (
                              <Typography.Text type={asset.grow ? 'success' : 'd
                              )}
                        </span>
                    </List.Item>
                    )}
                />
            </Card>
            ))}
        </Layout.Sider>
    );
};
```

frontend\src\components\layout\styles\AppContent.css

```
.content {
  text-align: center;
  height: calc(100vh - 128px);
  color: "#fff";
  background-color: #ffffff;
  padding: 2rem 1rem;
  overflow: hidden;
  display: flex;
  flex-direction: column;
  justify-content: space-between;
  gap: 2rem;
}

.title {
  text-align: left;
  color: #646464;
  text-shadow: 2px 2px 10px rgba(0, 0, 0, 0.2);
}
```

frontend\src\components\layout\styles\AppHeader.css

```
.header {
    width: 100%;
    height: fit-content;
    text-align: center;
    padding: 2rem 1rem;
    display: flex;
    justify-content: space-between;
    align-items: center;
    background-color: #FFFFFF;
    border-bottom: 1px solid #F0F0F0;
}

.wrapper {
    display: flex;
    align-items: center;
    gap: 1rem;
}

.logo {
    width: 4rem;
}

.title {
    font-size: 2.5rem;
    color: #132F49;
}
```

frontend\src\components\layout\styles\AppSider.css

```
.sider {
    height: calc(100vh - 128px);
    overflow-y: scroll;
    padding: 1rem;
    background-color: #ffffff;
    border-right: 1px solid #F0F0F0;
}

.sider::-webkit-scrollbar {
    display: none;
}
```

frontend\src\components\PortfolioChart.jsx

```
import { Chart as ChartJS, ArcElement, Tooltip, Legend, layouts } from "chart.js
import { Pie } from "react-chartjs-2";
import { useCrypto } from "../context/crypto-context";

ChartJS.register(ArcElement, Tooltip, Legend);

export default function PortfolioChart() {
  const { assets } = useCrypto();

  const data = {
    labels: assets.map((a) => a.name),
    datasets: [
      {
        label: "$",
        data: assets.map((a) => a.totalAmount),
        backgroundColor: [
          "rgba(19, 47, 73, 0.9)",
          "rgba(46, 139, 146, 0.9)",
          "rgba(108, 122, 137, 0.9)",
          "rgba(27, 79, 114, 0.9)",
          "rgba(75, 92, 107, 0.9)",
          "rgba(213, 216, 220, 0.9)",
          "rgba(22, 105, 122, 0.9)",
        ],
        borderColor: [
          "rgba(19, 47, 73, 1)",
          "rgba(46, 139, 146, 1)",
          "rgba(108, 122, 137, 1)",
          "rgba(27, 79, 114, 1)",
          "rgba(75, 92, 107, 1)",
          "rgba(213, 216, 220, 1)",
          "rgba(22, 105, 122, 1)",
        ],
        borderWidth: 1,
      },
    ],
  };

  const options = {
    maintainAspectRatio: false,
    responsive: true,
    plugins: {
      legend: {
        position: "left",
        labels: {
          padding: 20,
          font: {
            size: 16,
            weight: 600,
          },
        },
      },
    },
  };

  return (
    <div
      style={{
        display: "flex",
        marginBottom: "1rem",
        justifyContent: "center",
        height: 400,
      }}
    >
      <Pie data={data} options={options} />
    </div>
  );
}
```

frontend\src\components\styles\AssetsTable.css

```
.table {
    overflow-y: auto;
}

.table::-webkit-scrollbar {
    display: none;
}
```

frontend\src\context\crypto-context.jsx

```
import { createContext, useContext, useEffect, useState } from "react";
import { theme } from 'antd';
import { fakeFetchCrypto, fetchAssets } from '../api';
import { percentDifference } from "../utils";

const CryptoContext = createContext({
    assets: [],
    crypto: [],
    loading: false,
})

export function CryptoContextProvider({children}) {
    const [loading, setLoading] = useState(false)
    const [crypto, setCrypto] = useState([])
    const [assets, setAssets] = useState([])

    function mapAssets(assets, result) {
        return assets.map(asset => {
            const coin = result.find((c) => c.id === asset.id)
            return {
                grow: asset.price < coin.price,
                growPercent: percentDifference(asset.price, coin.price),
                totalAmount: asset.amount * coin.price,
                totalProfit: asset.amount * coin.price - asset.amount * asset.pr
                name: coin.name,
                ...asset,
            }
        })

    }

    useEffect(() => {
        async function preload() {
            setLoading(true);
            const { result } = await fakeFetchCrypto();
            const assets = await fetchAssets();

            setAssets(mapAssets(assets, result));
            setCrypto(result);
            setLoading(false);
        };
        preload()
    }, []);

    function addAsset(newAsset) {
        setAssets((prev) => mapAssets([...prev, newAsset], crypto))
    }

    return (
        <CryptoContext.Provider value={{loading, crypto, assets, addAsset}}>
            {children}
        </CryptoContext.Provider>
    )
}

export default CryptoContext

export function useCrypto() {
    return useContext(CryptoContext)
}
```

frontend\src\data.js

```
export const cryptoData = {
  result: [
    {
      id: 'bitcoin',
      icon: 'https://static.coinstats.app/coins/1650455588819.png',
      name: 'Bitcoin',
      symbol: 'BTC',
      rank: 1,
      price: 44870.39834657236,
      priceBtc: 1,
      volume: 35728788775.15447,
      marketCap: 879141227764.5575,
      availableSupply: 19592900,
      totalSupply: 21000000,
      priceChange1h: -0.34,
      priceChange1d: 0.94,
      priceChange1w: 5.02,
      redditUrl: 'https://www.reddit.com/r/Bitcoin/',
      websiteUrl: 'http://www.bitcoin.org',
      twitterUrl: 'https://twitter.com/bitcoin',
      explorers: [
        'https://blockchair.com/bitcoin/',
        'https://btc.com/',
        'https://btc.tokenview.io/',
        'https://www.oklink.com/btc',
        'https://3xpl.com/bitcoin',
        'https://blockchain.coinmarketcap.com/chain/bitcoin',
        'https://blockexplorer.one/btc/mainnet',
      ],
    },
    {
      id: 'ethereum',
      icon: 'https://static.coinstats.app/coins/1650455629727.png',
      name: 'Ethereum',
      symbol: 'ETH',
      rank: 2,
      price: 2262.9329473372445,
      priceBtc: 0.05043730802649077,
      volume: 15049137392.625889,
      marketCap: 271968761037.10645,
      availableSupply: 120184189,
      totalSupply: 120184189,
      priceChange1h: -0.07,
      priceChange1d: 0.42,
      priceChange1w: -1.87,
      redditUrl: 'https://www.reddit.com/r/ethereum',
      websiteUrl: 'https://www.ethereum.org/',
      twitterUrl: 'https://twitter.com/ethereum',
      contractAddress: '0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2',
      decimals: 18,
      explorers: [
        'https://etherscan.io/',
        'https://ethplorer.io/',
        'https://blockchair.com/ethereum',
        'https://eth.tokenview.io/',
        'https://www.oklink.com/eth',
        'https://3xpl.com/ethereum',
        'https://blockchain.coinmarketcap.com/chain/ethereum',
      ],
    },
    {
      id: 'tether',
      icon: 'https://static.coinstats.app/coins/1650455771843.png',
      name: 'Tether',
      symbol: 'USDT',
      rank: 3,
      price: 0.9958245177734858,
      priceBtc: 0.000022195402653l6914,
      volume: 40587650892.897285,
      marketCap: 93233699508.00995,
      availableSupply: 93624627476,
      totalSupply: 93624627476,
      priceChange1h: -0.22,
      priceChange1d: 0.02,
      priceChange1w: -0.2,
      redditUrl: 'https://www.reddit.com',
      websiteUrl: 'https://tether.to/',
      twitterUrl: 'https://twitter.com/Tether_to',
      contractAddress: '0xdac17f958d2ee523a2206206994597c13d831ec7',
      decimals: 18,
      explorers: [
        'https://etherscan.io/token/0xdac17f958d2ee523a2206206994597c13d831ec7',
        'https://ethplorer.io/address/0xdac17f958d2ee523a2206206994597c13d831ec7
        'https://bscscan.com/token/0x55d398326f99059ff775485246999027b3197955',
        'https://solscan.io/token/Es9vMFrzaCERmJfrF4H2FYD4KCoNkY11McCe8BenwNYB',
        'https://polygonscan.com/token/0xc2132d05d31c914a87c6611c10748aeb04b58e8
        'https://snowtrace.io/token/0x9702230a8ea53601f5cd2dc00fdbc13d4df4a8c7',
        'https://evmexplorer.velas.com/token/0xb44a9b6905af7c801311e8f4e76932ee9
        'https://avascan.info/blockchain/c/address/0x9702230a8ea53601f5cd2dc00fd
        'https://arbiscan.io/token/0xfd086bc7cd5c481dcc9c85ebe478a1c0b69fcbb9',
        'https://blockchair.com/ethereum/erc-20/token/0xdac17f958d2ee523a2206206
        'https://www.omniexplorer.info/asset/31',
      ],
    },
    {
      id: 'binance-coin',
      icon: 'https://static.coinstats.app/coins/1666608145347.png',
      name: 'BNB',
      symbol: 'BNB',
      rank: 4,
      price: 301.59863198670274,
      priceBtc: 0.006722171383726176,
      volume: 1632019815.3942976,
      marketCap: 46402804362.74094,
      availableSupply: 153856150,
      totalSupply: 153856150,
      priceChange1h: 0.09,
      priceChange1d: -1.98,
      priceChange1w: -2.93,
      redditUrl: 'https://www.reddit.com/r/binance',
      websiteUrl: 'https://www.binance.com',
      twitterUrl: 'https://twitter.com/binance',
      contractAddress: 'BNB',
      decimals: 18,
      explorers: [
        'https://bscscan.com',
        'https://explorer.binance.org/',
        'https://binance.mintscan.io/',
        'https://etherscan.io/token/0xb8c77482e45f1f44de1745f52c74426c631bdd52',
        'https://ethplorer.io/address/0xb8c77482e45f1f44de1745f52c74426c631bdd52
        'https://www.oklink.com/bsc',
        'https://3xpl.com/bnb',
        'https://explorer.energi.network/token/0xc3c19ee91cf3c1f7fbf3716a09d21dc
        'https://etherscan.io/token/0xB8c77482e45F1F44dE1745F52C74426C631bDD52',
      ],
    },
    {
      id: 'solana',
      icon: 'https://static.coinstats.app/coins/1701234596791.png',
      name: 'Solana',
      symbol: 'SOL',
      rank: 5,
      price: 93.7035752645126,
      priceBtc: 0.0020885091157300402,
      volume: 3591253070.807396,
      marketCap: 40461063524.96437,
      availableSupply: 431798503,
      totalSupply: 566552781,
      priceChange1h: -0.98,
      priceChange1d: -2.94,
      priceChange1w: -11.17,
      redditUrl: 'https://www.reddit.com/r/solana',
      websiteUrl: 'https://solana.com/',
      twitterUrl: 'https://twitter.com/solana',
      contractAddress: '0x7dff46370e9ea5f0bad3c4e29711ad50062ea7a4',
      decimals: 18,
      explorers: [
        'https://solscan.io/',
        'https://explorer.solana.com/',
        'https://solanabeach.io/',
        'https://solana.fm/',
        'https://www.oklink.com/sol',
      ],
    },
    {
      id: 'ripple',
      icon: 'https://static.coinstats.app/coins/XRPdnqGJ.png',
      name: 'XRP',
      symbol: 'XRP',
      rank: 6,
      price: 0.5629434018448091,
      priceBtc: 0.000012547145859419824,
      volume: 1103177305.849332,
      marketCap: 30517408090.486538,
      availableSupply: 54210437480,
      totalSupply: 99988065643,
      priceChange1h: -0.6,
      priceChange1d: -1.22,
      priceChange1w: -9.34,
      redditUrl: 'https://www.reddit.com/r/ripple',
      websiteUrl: 'https://ripple.com/currency/',
      twitterUrl: 'https://twitter.com/Ripple',
      contractAddress: '0x1d2f0da169ceb9fc7b3144628db156f3f6c60dbe',
      decimals: 18,
      explorers: [
        'https://blockchair.com/ripple',
        'https://xrpcharts.ripple.com',
        'https://xrpscan.com/',
        'https://bithomp.com/explorer/',
        'https://xrpcharts.ripple.com/#/graph/',
      ],
    },
    {
      id: 'usd-coin',
      icon: 'https://static.coinstats.app/coins/1650455825065.png',
      name: 'USDC',
      symbol: 'USDC',
      rank: 7,
      price: 1,
      priceBtc: 0.000022259755933599935,
      volume: 8964820403,
      marketCap: 25343872213,
      availableSupply: 25341702291,
      totalSupply: 25346562356,
      priceChange1h: 0.02,
      priceChange1d: -0.05,
      priceChange1w: -0.03,
      redditUrl: 'https://www.reddit.com',
      websiteUrl: 'https://www.circle.com/en/usdc',
      twitterUrl: 'https://twitter.com/circle',
      contractAddress: '0xa0b86991c6218b36c1d19d4a2e9eb0ce3606eb48',
      decimals: 18,
      explorers: [
        'https://etherscan.io/token/0xa0b86991c6218b36c1d19d4a2e9eb0ce3606eb48',
        'https://stepscan.io/token/0xe3f5a90f9cb311505cd691a46596599aa1a0ad7d',
        'https://nearblocks.io/token/17208628f84f5d6ad33f0da3bbbeb27ffcb398eac50
        'https://ethplorer.io/address/0xa0b86991c6218b36c1d19d4a2e9eb0ce3606eb48
        'https://basescan.org/token/0x833589fcd6edb6e08f4c7c32d4f71b54bda02913',
        'https://arbiscan.io/token/0xaf88d065e77c8cc2239327c5edb3a432268e5831',
        'https://hashscan.io/mainnet/token/0x00000000000000000000000000000000000
        'https://bscscan.com/token/0x8ac76a51cc950d9822d68b83fe1ad97b32cd580d',
        'https://www.oklink.com/en/okc/token/0xc946daf81b08146b1c7a8da2a851ddf2b
        'https://www.teloscan.io/token/0x818ec0a7fe18ff94269904fced6ae3dae6d6dc0
      ],
    },
    {
      id: 'staked-ether',
      icon: 'https://static.coinstats.app/coins/staked-etheruqt.png',
      name: 'Lido Staked Ether',
      symbol: 'STETH',
      rank: 8,
      price: 2263.48,
      priceBtc: 0.05038451236058478,
      volume: 12583680,
      marketCap: 20949523852,
      availableSupply: 9253636,
      totalSupply: 9253636,
      priceChange1h: -0.06,
      priceChange1d: 0.54,
      priceChange1w: -1.67,
      redditUrl: 'https://www.reddit.com/r/lidofinance/',
      websiteUrl: 'https://www.lido.fi',
      twitterUrl: 'https://twitter.com/lidofinance',
      contractAddress: '0xae7ab96520de3a18e5e111b5eaab095312d7fe84',
      decimals: 18,
      explorers: [
        'https://etherscan.io/token/0xae7ab96520de3a18e5e111b5eaab095312d7fe84',
        'https://ethplorer.io/address/0xae7ab96520de3a18e5e111b5eaab095312d7fe84
        'https://ethereum.dex.guru/token/0xae7ab96520de3a18e5e111b5eaab095312d7f
      ],
    },
    {
      id: 'cardano',
      icon: 'https://static.coinstats.app/coins/CardanojXddT.png',
      name: 'Cardano',
      symbol: 'ADA',
      rank: 9,
      price: 0.5071342431985936,
      priceBtc: 0.00001130324522654837,
      volume: 519686825.9246739,
```

frontend\src\index.css

```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
```

frontend\src\main.jsx

```
import { createRoot } from 'react-dom/client'
import App from './App.jsx'
import './index.css'

const root = document.getElementById('root')

createRoot(root).render(<App />)
```

frontend\src\utils.js

```
export function percentDifference(a ,b) {
    return +(100 * Math.abs((a - b) / ((a + b) / 2))).toFixed(2)
}

export function capitalize(str) {
    return str.charAt(0).toUpperCase() + str.substr(1)
}
```

frontend\vite.config.js

```
import { defineConfig } from 'vite'
import react from '@vitejs/plugin-react'

// https://vitejs.dev/config/
export default defineConfig({
  plugins: [react()],
})
```

package.json

```
{
  "name": "crypto-wallet",
  "version": "1.0.0",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node server.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": "",
  "dependencies": {
    "express": "^5.1.0"
  }
}
```

server.js

```
const express = require('express')
const app = express()
const port = 80

app.use(express.static('frontend/dist'))

app.listen(port, () => console.log('Server has been started on port 80 ...'))
```

##Обнаруженные зависимости

frontend\index.html

- /logo.svg
- /src/main.jsx

frontend\src\api.js

- ./data

frontend\vite.config.js

- vite
- @vitejs/plugin-react

server.js

- express