

REACT NATIVE амирхан

Салауаттылық және сауаттылық трекері

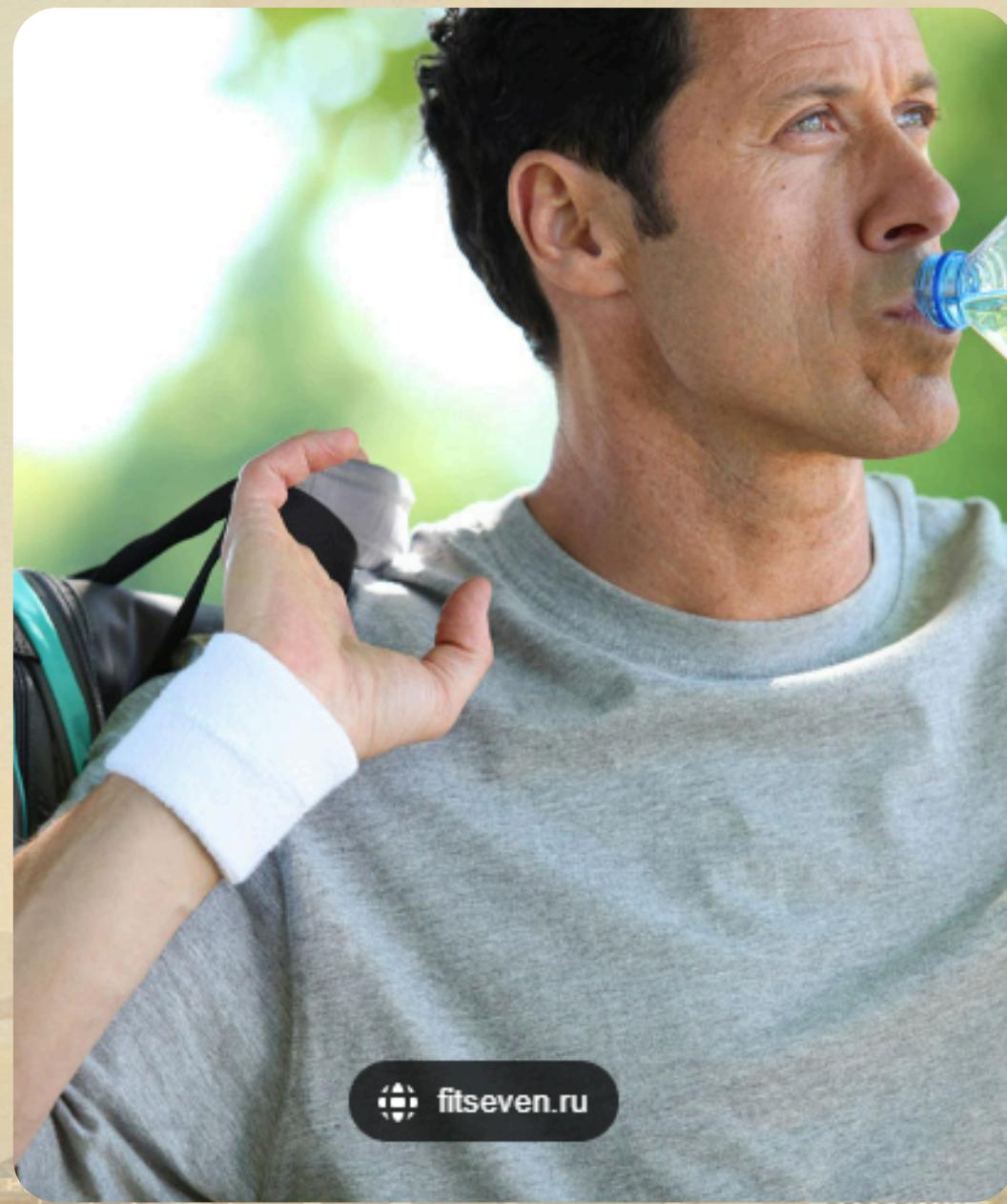
Орындаған: Темиров
Амирхан

КІРІСПЕ

Бұл бөлімде біз **React Native** көмегімен ауқымды жоба қосымшасын жасап шығарамыз. Бұл ерекше қосымша екеніне сенімдімін. Барынша мүмкіндіктер заманауи түрде жалғасын табатын болады. Бұл салауаттылық трекері. Өмір салтын ұстану, дұрыс тاماқтану, өз салмақ және бойыңызға қарай ұсынылатын жаттығылар кеңес ретінде берілетін болады, ақша бұл салауатты дұрыс басқаруға негізгі факторлардың бірі болғандықтан қаржы басқару мүмкіндігі тиісінше жақсы көмек бола алар терезе бар. Басында тіркелу бөлімі, сіз тіркелесіз, солай өз профиліңізді жасайсыз. Келесі негізі бетке көшесіз, ал ол жерде бұл қосымшаның негізгі мүмкіндіктерін айрықша көретін боласыз. Мүмкіндіктер көп болады, сол үшін қолданушылар көп болары анық, ал бұл қосымшаны одан дамыту үшін ынта берер еді. Қосымшаны Плэй-маркет қосымшасыннан жүктеп алсаныз болады. Сол арқылы менің мақтауыма сенетіңізге сенімім мол.

Жобаның мақсаты

Қолданушылардың денсаулығын
бірқалыпта ұстая
Қоғамда адамның жұмыс істеу үлесі
көптігіне байланысты уақыт тапшылығы
өз денсаулығына күтім жасауды
назардан алшақ ұстайды. Бұл әрине
қоғамның мәселесі болғандықтан бұл
қосымша салауаттылыққа назар салып
белгілеп, зерттеп, денсаулығын күтіпге
алуға көмектеседі.



Қажеттілігі

ЫНТАНЫ АРТТЫРУ

Еңгізген нәтижелеріне қарай өз
денсаулығының маңызды
бөлігін назарда ұстайды. Бұл
әрине жақсы және тұрақты
қолданушының әрекетіне орай
денсаулығының орташа
нәтижесін көріп ынта жігерін
одан әрі ашары сөзсіз.

УАҚЫТТЫ ТИІМДІ БАСҚАРА АЛУҒА НЕГІЗГІ МҮМКІНДІК

Қосымшаның өнімділігінің
жұмысы жылдам әрі заманға
сай сапалы болғандықтан және
сәнді болғандықтан қолдануға
өте ыңғайлы әрі жұмысыннан
босаған уақытта уақытың
салауаттылыққа арнауға
мүмкіндік береді..

Жоба жұмысының барысы *App.js*

1. Қосымшаның жұмысы:

Бұл қосымша бірнеше кітапханармен жұмыс жасатамыз. Бастау реті үшін бұл кітапханалардың маңызы зор.

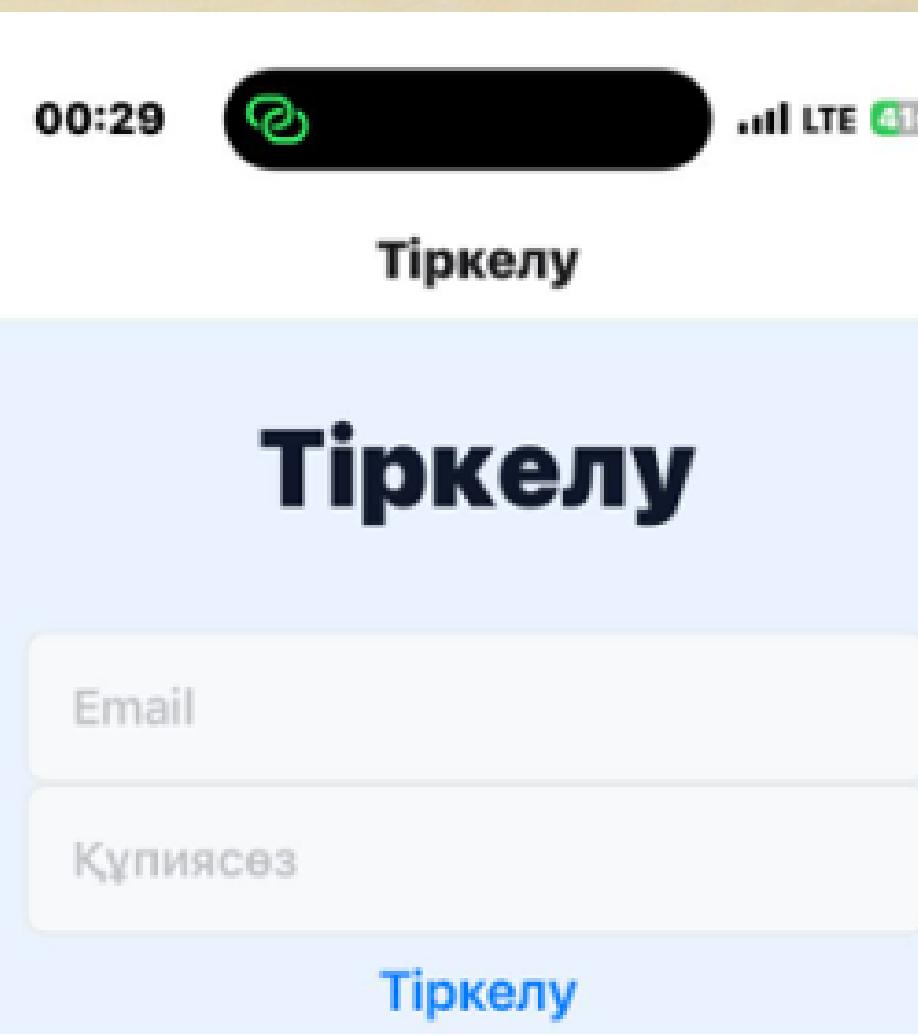
2. Бізде Топ навигация функциясы.

Бұл біз пайдаланатын экрандардың кіру мүмкіндігінің жұмысына жауап беретін болады. Текст стилі арқылы кнопкалардың көрінісінде көрсетіп, модель жұмысының жолына дайындаپ беретіне болады.

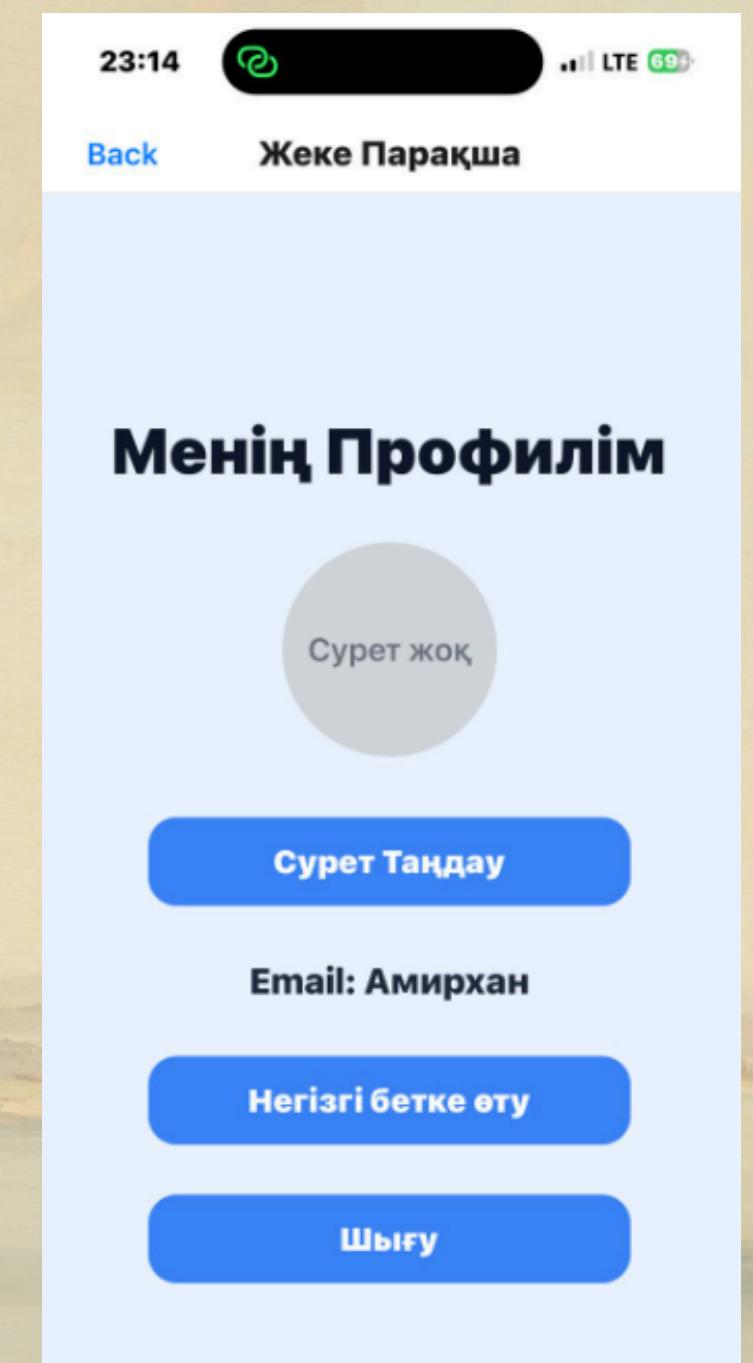
3. Бұл регистрация мүмкіндігін жасау жолы. **UseEffect** арқылы навигация жолын шартпен жүргізіп отырады. Бұл навигация арқылы Профиль және Хом терзесімен байланыс орнатады.

Профиль

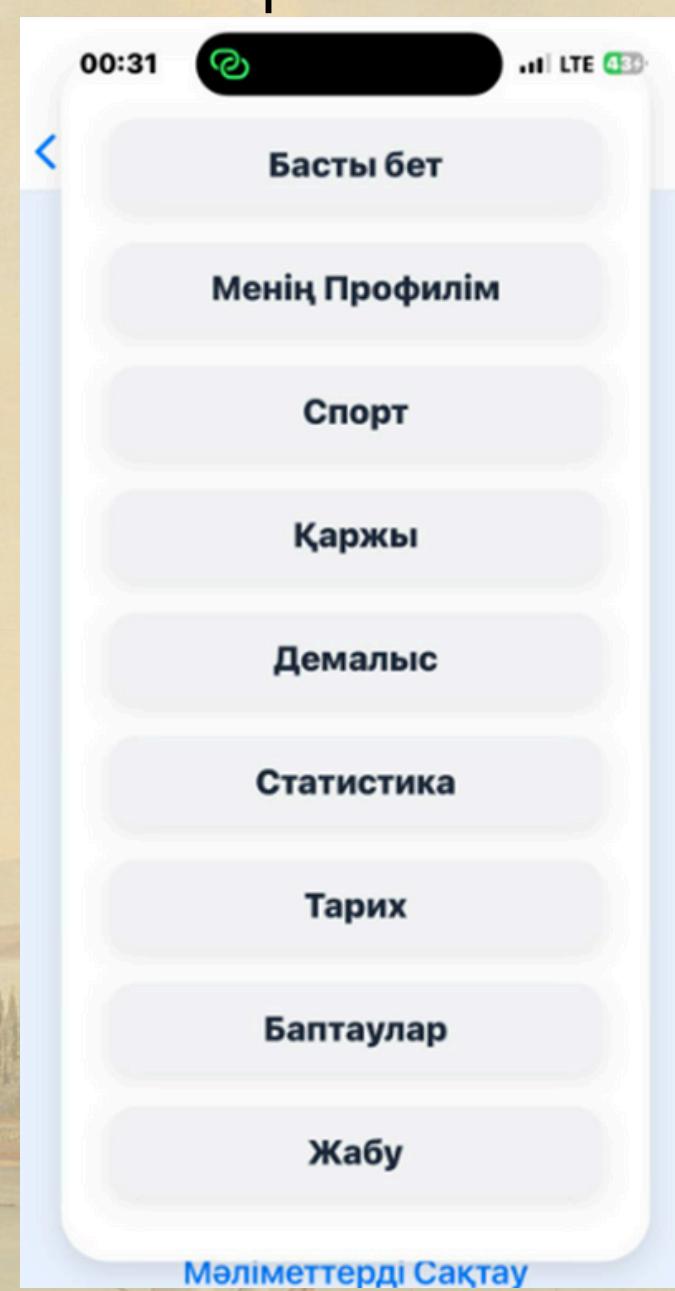
Мұнда қолданушы өз атын
жазып құпиясөз жазып
тіркелу бөліміннен өту тиң



Профиль жасалынады.



Бұл жерде барлығы
байланысқаның көрсететін
экран шығаратын жолақша



Тіркелу

БҰЛ КОД ФРАГМЕНТИНДЕ РЕГСИТРАЦИЯ ТЕРЕЗЕҢІҢ МАҢЫЗДЫ ЖҰМЫСЫ БАР. КОДТА **ASYNC** ЖӘНЕ **HANDLE REGISTER** ФУНКЦИЯЛАРЫНДА ҚОЛДАНЫЛҒАН. БҰЛ ФУНКЦИЯЛАР **AWAIT** КІЛТТІК СӨЗІН ҚОЛДАНАН ОТЫРЫП, АСИНХРОНДЫ ӘРЕКЕТТЕРДІ КҮТІП, ОРЫНДАУ КЕЗІНДЕ КОДТЫҢ ОРЫНДАЛУЫН БӨГЕМЕЙДІ.

```
53 const RegisterScreen = ({ navigation }) => {
54   const [email, setEmail] = useState('');
55   const [password, setPassword] = useState('');
56
57   useEffect(() => {
58     const checkUser = async () => {
59       const userData = await AsyncStorage.getItem('user');
60       if (userData) {
61         navigation.navigate('Profile');
62         navigation.navigate('Home');
63       }
64     };
65     checkUser();
66   }, []);
67
68
69   const handleRegister = async () => {
70     if (email && password) {
71       await AsyncStorage.setItem('user', JSON.stringify({ email, password }));
72       alert('Тіркелу сәтті аяқталды!');
73       navigation.navigate('Profile');
74       navigation.navigate('Home');
75       navigation.navigate('Profile');
76     } else {
77       alert('Барлық жолдарды толтырыңыз!');
78     }
79   };
80
81   return (
82     <View style={styles.container}>
83       <Text style={styles.header}>Тіркелу</Text>
84       <TextInput
85         style={styles.input}
86         placeholder="Email"
87         value={email}
88         onChangeText={setEmail}
89       />
```

негізгі терезе

КЕЛЕСІ БІЗІҢ БАСТЫ
ЭКРАНЫМЫЗДЫ ЖАСАП ТҰРҒАН
КОД ЖҰМЫСЫ. БҰЛ ЖЕРДЕ
БАРЛЫҒЫ БАЙЛАНЫСҚАНЫН
КӨРСЕТЕТИН ЭКРАН ШЫҒАРАТЫН
ЖОЛАҚША **BUTTON** КНОПКАСЫН
ҚОЛДАЙДЫ.
КОДТА **USEEFFECT** КОМПАНЕНТИ
АЛҒАШ РЕТ ЖҮКТЕЛГЕНДЕ
CHECKUSER ФУНКЦИЯСЫН
ШАҚЫРУ ҮШІН ҚОЛДАНЫЛАДЫ.

```
const HomeScreen = ({ navigation }) => {
  const [суIшу, суIшуОрнату] = useState(0);
  const [үйқыСағаты, үйқыСағатыОрнату] = useState('');
  const [көңілКүй, көңілКүйОрнату] = useState('');
  const [тарих, тарихОрнату] = useState([]);
  const [modalVisible, setModalVisible] = useState(false);

  useEffect(() => {
    мәліметтердіЖүктеу();
  }, []);

  const мәліметтердіЖүктеу = async () => {
    try {
      const сақталғанСу = await AsyncStorage.getItem('суIшу');
      const сақталғанҮйқы = await AsyncStorage.getItem('үйқыСағаты');
      const сақталғанТарих = await AsyncStorage.getItem('тарих');
      if (сақталғанСу) суIшуОрнату(parseInt(сақталғанСу));
      if (сақталғанҮйқы) үйқыСағатыОрнату(сақталғанҮйқы);
      if (сақталғанТарих) тарихОрнату(JSON.parse(сақталғанТарих));
    } catch {
      Alert.alert('Қате', 'Мәліметтерді жүктеу мүмкін болмады.');
    }
  };

  const мәліметтердіСақтау = async () => {
    try {
      const жаңаТарих = [
        ...тарих,
        {
          су: суIшу,
          үйқы: үйқыСағаты,
          көңілКүй,
          уақыт: new Date().toLocaleString(),
        },
      ];
      тарихОрнату(жаңаТарих);
      await AsyncStorage.setItem('тарих', JSON.stringify(жаңаТарих));
      await AsyncStorage.setItem('суIшу', суIшу.toString());
      await AsyncStorage.setItem('үйқыСағаты', үйқыСағаты);
      Alert.alert('Сәтті!', 'Мәліметтер сақталды!');
    } catch {
      Alert.alert('Қате', 'Мәліметтерді сақтау мүмкін болмады.');
    }
  };
}
```

Back

Салауаттылық

Amirkhan
Трекері



Су Iшу (Литр)

0

Үйқы Сағаты

мысалы, 7.5

Көңіл Күй

Көңіл күйінді жазыңыз

Мәліметтерді Сақтау

Профиль жасалынады. Бұл код фрагменті Хом терезесімен байланыста болады.

UseEffect ішінде **user** және **profilePhoto** жасалынып **savedPhoto**-да сақталып отырады.

```
const ProfileScreen = ({ navigation }) => {
  const [user, setUser] = useState(null);
  const [profilePhoto, setProfilePhoto] = useState(null);

  useEffect(() => {
    const loadUser = async () => {
      const userData = await AsyncStorage.getItem('user');
      setUser(JSON.parse(userData));
      const savedPhoto = await AsyncStorage.getItem('profilePhoto');
      if (savedPhoto) {
        setProfilePhoto(savedPhoto);
      }
    };
    loadUser();
  }, []);
```

Негізгі Хом терезесіннен кейін біз басты терезесінің мүмкіндіктерін қолданып бастау жолының жұмыс кодын жасадық. Бұл салмағыңыздың және бойыңыз бойынша есептейді калькулятор **if** шартпен жұмыс жасайтын болады. Бұл әрине формула арқылы жүзеге асады.

```
const SportScreen = () => {
  const [weight, setWeight] = useState('');
  const [height, setHeight] = useState('');
  const [calories, setCalories] = useState(null);
  const [bmi, setBmi] = useState(null);
  const [advice, setAdvice] = useState('');
  const [immunity, setImmunity] = useState(null);
  const [mealPlan, setMealPlan] = useState(null);
  const [exerciseAdvice, setExerciseAdvice] = useState([]);
  const [timeBasedExercise, setTimeBasedExercise] = useState([]);

  const calculateCalories = () => {
    if (weight && height) {
      const heightInMeters = parseFloat(height) / 100;
      const calculatedBmi = parseFloat(weight) / (heightInMeters * heightInMeters);
      const recommendedCalories = Math.round(2000 + (calculatedBmi - 22) * 100);
      setCalories(recommendedCalories);
      setBmi(calculatedBmi);
    }
  }

  const immunityLevel = Math.min(100, Math.max(0, Math.round((calculatedBmi / 25) * 100)));
  setImmunity(immunityLevel);

  const isOverweight = calculatedBmi > 24.9;
```

Салауаттылық Спорт

Спорт

Салмағыңыз (кг)

70

Бойыңыз (см)

180

Калорияны және иммунитетті есептеу

Сізге тәулігіне шамамен 1960 калория қажет.

Сіздің BMI: 21.6

Сіздің салмағыңыз нормада. Орташа қарқынмен жаттығу жасаңыз.

Уақытқа байланысты ұсынылатын жаттығулар:

Таңертең: Орташа қарқынды жаттығулар

Түсте: Салмақпен жаттығу

Кешке: Жұзу немесе созылу жаттығулары

Ұсынылатын тағамдар:

Салауаттылық Спорт

Ұсынылатын тағамдар:

Таңғы ас:

Жұмыртқа — 155 ккал,
Иммунитетті арттыру: 5%

Сұлы ботқасы — 100 ккал,
Иммунитетті арттыру: 5%

Жеміс-жидектер — 120 ккал,
Иммунитетті арттыру: 10%

Түскі ас:

Қайнатылған тауық еті — 180 ккал, Иммунитетті арттыру: 10%

Көкөніс салаты — 120 ккал,
Иммунитетті арттыру: 8%

Балық — 200 ккал,
Иммунитетті арттыру: 15%

Кешкі ас:

Сорпа (бүршақтан) — 150 ккал, Иммунитетті арттыру: 8%

Салауаттылық Спорт

Қайнатылған тауық еті — 180 ккал, Иммунитетті арттыру: 10%

Көкөніс салаты — 120 ккал,
Иммунитетті арттыру: 8%

Балық — 200 ккал,
Иммунитетті арттыру: 15%

Кешкі ас:

Сорпа (бүршақтан) — 150 ккал, Иммунитетті арттыру: 8%

Авокадо — 240 ккал,
Иммунитетті арттыру: 15%

Сүт өнімдері — 150 ккал,
Иммунитетті арттыру: 10%

Дене формасын жақсартуға арналған жалпы кеңестер:

1. Күнделікті жаттығу жасаңыз.
2. Тенгерімді тамақтануды ұстаныңыз.
3. Су ішуге көзіл беліңіз.



Салауаттылық Қаржы

Жалақыңыз (теңге)

400000

Шығындарыңыз (теңге)

200000

Мақсатыңыз қандай?
(мысалы, үй, машина)

Машина

Оның бағасы қандай? (теңге)

40000000

Қанша уақытқа алуды
жоспарлап жатырысыз? (ай)

12

```

504 const [goal, setGoal] = useState('');
505 const [goalPrice, setGoalPrice] = useState('');
506 const [time, setTime] = useState('');
507 const [advice, setAdvice] = useState(null);
508 const [chartData, setChartData] = useState({ income: 0, expenses: 0, savings: 0 });
509 const [expenseBreakdown, setExpenseBreakdown] = useState({ self: 0, food: 0, home: 0, clothes: 0 });
510 const [savingBreakdown, setSavingBreakdown] = useState({ self: 0, food: 0, home: 0, clothes: 0 });

511
512 const provideAdvice = () => {
513   if (salary && expenses && goal && goalPrice && time) {
514     const salaryNum = parseFloat(salary);
515     const expensesNum = parseFloat(expenses);
516     const savings = salaryNum - expensesNum; // Жинақ
517     const goalCost = parseFloat(goalPrice);
518     const goalTime = parseFloat(time);

519     setChartData({ income: salaryNum, expenses: expensesNum, savings });

520     // Шығындардың үлесін бөлу
521     const expenseTotal = expensesNum || 1; // Бөлуді қателіктерден сақтау үшін
522     const breakdown = {
523       self: parseFloat((expensesNum * 0.4).toFixed(2)),
524       food: parseFloat((expensesNum * 0.3).toFixed(2)),
525       home: parseFloat((expensesNum * 0.1).toFixed(2)),
526       clothes: parseFloat((expensesNum * 0.2).toFixed(2))
527     };
528     setExpenseBreakdown(breakdown);
529     setSavingBreakdown(breakdown);
530   }
531 }
532 
```

Келесі қаржы код бөлігі. Жинақ жасауды, шығындарду басқару мүмкіндігіне жауап береді. Қолданушы қажетті бөлімдерді енгизу керек. Сол арқылы нітижені тексеруге жібереді. Бұл тек біржақты шешімін алмай көп жақты өз жағдайын түсінуге артықшылық береді.

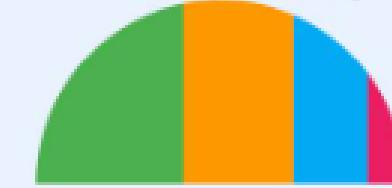
Кеңес алады және шешім іздейді, қолданушыға ой салатын болады.

Салауаттылық Қаржы

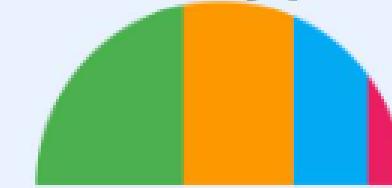
Кеңес Алу

Сіз "Машина" мақсатқа 12 айда жете алмайсыз. Егер ай сайын қосымша 3133333.33 теңге үнемдесеніз, 12 ай ішінде мақсатқа жете аласыз.

Шығындардың үлесі



Жинақтау үлесі



Өз қажеттілігі Тамак Үй
Киім

Ақшаны үнемдеуге арналған кеңестер:

Статистика

Орташа Су Ішу (Литр):

3.00

Орташа Ұйқы Сағаты:

7.00

Келесі біздің маңызға ие тағы терезелердің бірі ол біздің медитация тереземізде бар, ол таймер мүмкіндігі бойынша жұмыс жасайды.

Екінші біз енгізген ақпарат сақталатын тарих бөлімі, біз барлығын жоя аламыз.

Үшінші қоланушының нәтижелеріне сәйкес статистика бөлімі.

Back

Meditation

Демалыс

Қалған уақыт: 10 минут

[Медитацияны бастау](#)

[Қайта орнату](#)

Медитация жасауға арналған кеңестер:

1. Ұңғайлы жерде отырыңыз.
2. Терең дем алып, баяу дем шығарыңыз.
3. Ойыңызды тыныштандырыңыз.

Салауаттылық Тарих

Тарих

Су: 3 Л
Ұйқы: 7 сағат
Көніл күй: Бсдс
Уақыт: 15.12.2024, 01:00:57

Су: 3 Л
Ұйқы: 7 сағат
Көніл күй:
Уақыт: 15.12.2024, 01:09:31

[Тарихты Өшіру](#)

Настройка және орталық Көмек

Бізде хабарламаларды
жазуға қорғаныш қоя
аламыз.

Және тұнгілжұндізгі
режимге ауыстыра
аламыз.

19:42

Салауаттылық Көмек

Көмек

Сұрақтарыңыз болса
немесе көмек қажет
болса, бізге
хабарласыңыз:

Электронды пошта:
amirkhantemirov04@gmail.co
m

Негізгі бетке оралу

Back

Настройка

Настройка

Сендерілген

Қосулы

ҚОРЫТЫНДЫ

Қорытындылай келе бұл заманғай сай қосымшы. Әрине бұл
қосымшаның пайдасы заманнан ешқашанда қалмайды.

Себебі адам денсаулығы барлығыннанда мағызды
болғандықтан бұл қосымша қолданушыны бақылауға ала
алатын болады. Қосымша мүмкіндігі өте маңызды, әрі заман
дамып жатсада өз маңызын жоғалтпайды.

