
ДИСЦИПЛИНА	Фронтенд и бэкенд разработка
ИНСТИТУТ	ИПТИП
КАФЕДРА	Индустриального программирования
ВИД УЧЕБНОГО МАТЕРИАЛА	Методические указания к практическим занятиям
ПРЕПОДАВАТЕЛЬ	Астафьев Рустам Уралович
СЕМЕСТР	1 семестр, 2025/2026 уч. год

Ссылка на материал:

<https://github.com/astafiev-rustam/frontend-and-backend-development/tree/practice-1-26>

Практическое занятие 26: Использование UI-кит

В рамках данного занятия будут рассмотрены возможности работы с готовыми наборами компонентов и библиотек в приложениях React.

Теоретическая часть

Пример 1. Интеграция Material-UI (MUI)

Проблема: Нужно быстро создать профессиональный интерфейс без написания CSS с нуля.

Подход к решению: Используем Material-UI - популярную React UI-библиотеку с готовыми компонентами.

Установка и настройка:

```
npm install @mui/material @emotion/react @emotion/styled  
npm install @mui/icons-material # Иконки
```

Исходный код в файле SimpleTechCard.jsx:

```
import React from 'react';  
import {  
  Card,  
  CardContent,  
  CardActions,  
  Typography,  
  Button,  
  Chip,  
  Box  
} from '@mui/material';
```

```
function SimpleTechCard({ technology, onStatusChange }) {
  const getStatusColor = (status) => {
    switch (status) {
      case 'completed': return 'success';
      case 'in-progress': return 'warning';
      default: return 'default';
    }
  };

  const getStatusText = (status) => {
    switch (status) {
      case 'completed': return 'Завершено';
      case 'in-progress': return 'В процессе';
      default: return 'Не начато';
    }
  };
}

return (
  <Card sx={{ maxWidth: 345, margin: 2 }}>
    <CardContent>
      <Typography variant="h5" component="h2" gutterBottom>
        {technology.title}
      </Typography>

      <Typography variant="body2" color="text.secondary" sx={{ mb: 2 }}>
        {technology.description}
      </Typography>

      <Box sx={{ display: 'flex', gap: 1, flexWrap: 'wrap' }}>
        <Chip
          label={technology.category}
          variant="outlined"
          size="small"
        />
        <Chip
          label={getStatusText(technology.status)}
          color={getStatusColor(technology.status)}
          size="small"
        />
      </Box>
    </CardContent>

    <CardActions>
      {technology.status !== 'completed' && (
        <Button
          size="small"
          variant="contained"
          onClick={() => onStatusChange(technology.id, 'completed')}
        >
          Завершить
        </Button>
      )}
    <Button>
```

```
        size="small"
        variant="outlined"
        onClick={() => onStatusChange(technology.id,
            technology.status === 'in-progress' ? 'not-started' : 'in-progress')}
    >
    {technology.status === 'in-progress' ? 'Приостановить' : 'Начать'}
  </Button>
</CardActions>
</Card>
);
}

export default SimpleTechCard;
```

Изменим файл [App.js](#) для работы приложения:

```
import React, { useState } from 'react';
import {
  ThemeProvider,
  createTheme,
  Container,
  Typography,
  Button,
  Grid,
  Box,
  CssBaseline
} from '@mui/material';
import { Add } from '@mui/icons-material';
import SimpleTechCard from './SimpleTechCard';

// Создаем тему
const theme = createTheme({
  palette: {
    primary: {
      main: '#1976d2',
    },
    secondary: {
      main: '#dc004e',
    },
  },
});

function App() {
  const [technologies, setTechnologies] = useState([
    {
      id: 1,
      title: 'React Components',
      description: 'Изучение функциональных и классовых компонентов',
      category: 'frontend',
      status: 'in-progress'
    },
  ])
```

```
id: 2,
title: 'Material-UI',
description: 'Освоение Material Design для React',
category: 'ui-library',
status: 'not-started'
},
{
id: 3,
title: 'React Hooks',
description: 'Использование useState, useEffect и других хуков',
category: 'frontend',
status: 'completed'
}
]);
};

const handleStatusChange = (techId, newStatus) => {
setTechnologies(prev =>
prev.map(tech =>
tech.id === techId ? { ...tech, status: newStatus } : tech
)
);
};

const addNewTechnology = () => {
const newTech = {
id: Date.now(),
title: `Новая технология ${technologies.length + 1}`,
description: 'Описание новой технологии для изучения',
category: 'other',
status: 'not-started'
};
setTechnologies(prev => [...prev, newTech]);
};

return (
<ThemeProvider theme={theme}>
<CssBaseline />
<Container maxWidth="lg" sx={{ py: 4 }}>
/* Заголовок */
<Box sx={{ textAlign: 'center', mb: 4 }}>
<Typography variant="h3" component="h1" gutterBottom>
 Трекер технологий
</Typography>
<Typography variant="h6" color="text.secondary">
Отслеживайте прогресс изучения технологий
</Typography>
</Box>

/* Кнопка добавления */
<Box sx={{ textAlign: 'center', mb: 4 }}>
<Button
variant="contained"
startIcon={<Add />}
onClick={addNewTechnology}

```

```
        size="large"
      >
      Добавить технологию
    </Button>
  </Box>

  {/* Сетка карточек */}
  <Grid container spacing={3}>
    {technologies.map(technology => (
      <Grid item xs={12} sm={6} md={4} key={technology.id}>
        <SimpleTechCard
          technology={technology}
          onStatusChange={handleStatusChange}
        />
      </Grid>
    )));
  </Grid>

  {/* Статистика */}
  <Box sx={{ mt: 4, p: 2, backgroundColor: 'grey.50', borderRadius: 2 }}>
    <Typography variant="h6" gutterBottom>
      Статистика:
    </Typography>
    <Typography>
      Всего: {technologies.length} |
      Завершено: {technologies.filter(t => t.status === 'completed').length}
    | 
      В процессе: {technologies.filter(t => t.status === 'in-progress').length}
    </Typography>
  </Box>
</Container>
</ThemeProvider>
);
}

export default App;
```

Пример 2. Dashboard с различными UI-компонентами

Исходный код в файле `Dashboard.jsx`:

```
import React from 'react';
import {
  Box,
  Grid,
  Card,
 CardContent,
  Typography,
  LinearProgress,
  Chip,
```

```
Paper,
List,
ListItem,
ListItemText,
ListItemIcon,
Divider,
AppBar,
Toolbar,
IconButton,
Badge,
Tabs,
Tab

} from '@mui/material';
import {
  Notifications as NotificationsIcon,
  AccountCircle as AccountCircleIcon,
  CheckCircle as CheckCircleIcon,
  Schedule as ScheduleIcon,
  RadioButtonUnchecked as RadioButtonUncheckedIcon
} from '@mui/icons-material';

function TabPanel({ children, value, index, ...other }) {
  return (
    <div
      role="tabpanel"
      hidden={value !== index}
      id={`dashboard-tabpanel-${index}`}
      aria-labelledby={`dashboard-tab-${index}`}
      {...other}
    >
      {value === index && <Box sx={{ p: 3 }}>{children}</Box>}
    </div>
  );
}

function Dashboard({ technologies }) {
  const [tabValue, setTabValue] = React.useState(0);
  const [notificationCount] = React.useState(3);

  // Статистика
  const stats = {
    total: technologies.length,
    completed: technologies.filter(t => t.status === 'completed').length,
    inProgress: technologies.filter(t => t.status === 'in-progress').length,
    notStarted: technologies.filter(t => t.status === 'not-started').length,
    progress: technologies.length > 0 ?
      Math.round((technologies.filter(t => t.status === 'completed').length /
      technologies.length) * 100) : 0
  };

  // Предстоящие дедлайны (если бы они были)
  const upcomingDeadlines = technologies
    .filter(t => t.deadline && t.status !== 'completed')
    .sort((a, b) => new Date(a.deadline) - new Date(b.deadline))
}
```

```
.slice(0, 5);

// Недавно добавленные
const recentTechnologies = technologies
  .sort((a, b) => new Date(b.createdAt || 0) - new Date(a.createdAt || 0))
  .slice(0, 5);

const handleTabChange = (event, newValue) => {
  setTabValue(newValue);
};

return (
  <Box sx={{ flexGrow: 1 }}>
    {/* Шапка */}
    <AppBar position="static" color="default" elevation={1}>
      <Toolbar>
        <Typography variant="h6" component="div" sx={{ flexGrow: 1 }}>
           Панель управления
        </Typography>

        <IconButton color="inherit">
          <Badge badgeContent={notificationCount} color="error">
            <NotificationsIcon />
          </Badge>
        </IconButton>

        <IconButton color="inherit">
          <AccountCircleIcon />
        </IconButton>
      </Toolbar>
    </AppBar>

    {/* Табы */}
    <Paper sx={{ borderBottom: 1, borderColor: 'divider' }}>
      <Tabs value={tabValue} onChange={handleTabChange} aria-label="dashboard
      tabs">
        <Tab label="Обзор" />
        <Tab label="Статистика" />
        <Tab label="Активность" />
      </Tabs>
    </Paper>

    {/* Вкладка обзора */}
    <TabPanel value={tabValue} index={0}>
      <Grid container spacing={3}>
        {/* Статистические карточки */}
        <Grid item xs={12} sm={6} md={3}>
          <Card>
            <CardContent>
              <Typography color="text.secondary" gutterBottom>
                Всего технологий
              </Typography>
              <Typography variant="h4" component="div">
                {stats.total}
              </Typography>
            </CardContent>
          </Card>
        </Grid item>
      </Grid>
    </TabPanel>
  </Box>
)
```

```
        </Typography>
    </CardContent>
</Card>
</Grid>

<Grid item xs={12} sm={6} md={3}>
    <Card>
        <CardContent>
            <Typography color="text.secondary" gutterBottom>
                Завершено
            </Typography>
            <Typography variant="h4" component="div" color="success.main">
                {stats.completed}
            </Typography>
        </CardContent>
    </Card>
</Grid>

<Grid item xs={12} sm={6} md={3}>
    <Card>
        <CardContent>
            <Typography color="text.secondary" gutterBottom>
                В процессе
            </Typography>
            <Typography variant="h4" component="div" color="warning.main">
                {stats.inProgress}
            </Typography>
        </CardContent>
    </Card>
</Grid>

<Grid item xs={12} sm={6} md={3}>
    <Card>
        <CardContent>
            <Typography color="text.secondary" gutterBottom>
                Не начато
            </Typography>
            <Typography variant="h4" component="div" color="text.secondary">
                {stats.notStarted}
            </Typography>
        </CardContent>
    </Card>
</Grid>

/* Прогресс */
<Grid item xs={12}>
    <Card>
        <CardContent>
            <Typography variant="h6" gutterBottom>
                Общий прогресс
            </Typography>
            <Box display="flex" alignItems="center" gap={2}>
                <Box flex={1}>
                    <LinearProgress
```

```
        variant="determinate"
        value={stats.progress}
        sx={{ height: 10, borderRadius: 5 }}
      />
    </Box>
    <Typography variant="h6" color="primary">
      {stats.progress}%
    </Typography>
  </Box>
</CardContent>
</Card>
</Grid>

/* Предстоящие дедлайны */
<Grid item xs={12} md={6}>
  <Card>
    <CardContent>
      <Typography variant="h6" gutterBottom>
        📅 Активные технологии
      </Typography>
      <List>
        {technologies.filter(t => t.status === 'in-progress').map((tech,
index) => (
          <React.Fragment key={tech.id}>
            <ListItem>
              <ListItemIcon>
                <ScheduleIcon color="warning" />
              </ListItemIcon>
              <ListItemText
                primary={tech.title}
                secondary={tech.category}
              />
              <Chip
                label="В процессе"
                size="small"
                color="warning"
              />
            </ListItem>
            {index < technologies.filter(t => t.status === 'in-
progress').length - 1 && <Divider />}
          </React.Fragment>
        )))
      
```

{technologies.filter(t => t.status === 'in-progress').length === 0 && (

```
        <ListItem>
          <ListItemText
            primary="Нет активных технологий"
            secondary="Начните изучение новой технологии"
          />
        </ListItem>
      )})
    </List>
  </CardContent>
```

```
</Card>
</Grid>

{/* Недавно добавленные */}
<Grid item xs={12} md={6}>
  <Card>
    <CardContent>
      <Typography variant="h6" gutterBottom>
        NEW Недавно добавленные
      </Typography>
      <List>
        {recentTechnologies.map((tech, index) => (
          <React.Fragment key={tech.id}>
            <ListItem>
              <ListItemIcon>
                {tech.status === 'completed' ? (
                  <CheckCircleIcon color="success" />
                ) : tech.status === 'in-progress' ? (
                  <ScheduleIcon color="warning" />
                ) : (
                  <RadioButtonUncheckedIcon color="disabled" />
                )}
              </ListItemIcon>
              <ListItemText primary={tech.title} secondary={tech.category} />
            <Chip
              label={tech.status === 'completed' ? 'Завершено' :
                tech.status === 'in-progress' ? 'В процессе' :
                'Начато'}
              size="small"
              variant="outlined"
            />
          </React.Fragment>
        )))
      </List>
    <CardContent>
  </Card>
</Grid>
</Grid>
</TabPanel>

{/* Вкладка статистики */}
<TabPanel value={tabValue} index={1}>
  <Typography variant="h4" gutterBottom>
    Детальная статистика
  </Typography>

  <Grid container spacing={3}>
    <Grid item xs={12} md={6}>
      <Card>
```

```
<CardContent>
  <Typography variant="h6" gutterBottom>
    Распределение по статусам
  </Typography>
  <Box sx={{ mt: 2 }}>
    <Box display="flex" justifyContent="space-between" mb={1}>
      <Typography>Завершено</Typography>
      <Typography>{stats.completed} ({stats.progress}%)</Typography>
    </Box>
    <LinearProgress variant="determinate" value={stats.progress}
color="success" />

    <Box display="flex" justifyContent="space-between" mt={2} mb={1}>
      <Typography>В процессе</Typography>
      <Typography>{stats.inProgress}</Typography>
    </Box>
    <LinearProgress
      variant="determinate"
      value={(stats.inProgress / stats.total) * 100}
      color="warning"
    />

    <Box display="flex" justifyContent="space-between" mt={2} mb={1}>
      <Typography>Не начато</Typography>
      <Typography>{stats.notStarted}</Typography>
    </Box>
    <LinearProgress
      variant="determinate"
      value={(stats.notStarted / stats.total) * 100}
      color="inherit"
    />
  </Box>
  </CardContent>
</Card>
</Grid>
</Grid>
</TabPanel>

{/* Вкладка активности */}
<TabPanel value={tabValue} index={2}>
  <Typography variant="h4" gutterBottom>
    История активности
  </Typography>
  <Typography color="text.secondary">
    Здесь будет отображаться история изменений статусов...
  </Typography>
</TabPanel>
</Box>
);

}

export default Dashboard;
```

Обновим `App.js`:

```
import React, { useState } from 'react';
import {
  ThemeProvider,
  createTheme,
  Container,
  Typography,
  Button,
  Grid,
  Box,
  CssBaseline,
  AppBar,
  Toolbar,
  Tabs,
  Tab
} from '@mui/material';
import { Add, Dashboard as DashboardIcon, List as ListIcon } from '@mui/icons-material';
import SimpleTechCard from './SimpleTechCard';
import Dashboard from './Dashboard';

const theme = createTheme({
  palette: {
    primary: {
      main: '#1976d2',
    },
    secondary: {
      main: '#dc004e',
    },
  },
});

function TabPanel({ children, value, index, ...other }) {
  return (
    <div
      role="tabpanel"
      hidden={value !== index}
      id={`app-tabpanel-${index}`}
      aria-labelledby={`app-tab-${index}`}
      {...other}
    >
      {value === index && <Box sx={{ p: 3 }}>{children}</Box>}
    </div>
  );
}

function App() {
  const [tabValue, setTabValue] = useState(0);
  const [technologies, setTechnologies] = useState([
    {
      title: 'React',
      icon: Add,
      description: 'A JavaScript library for building user interfaces. It provides a declarative, component-based, and efficient way to manage UI state and logic. React is used by millions of developers worldwide and is the foundation for many popular web and mobile applications.',
      link: 'https://reactjs.org'
    },
    {
      title: 'Node.js',
      icon: DashboardIcon,
      description: 'A JavaScript runtime built on V8 that executes JavaScript code outside a browser. It is used for building server-side web applications and various other server-side tasks. Node.js uses the same engine as Google's Chrome browser, making it fast and efficient for handling real-time data processing and streaming.',
      link: 'https://nodejs.org'
    },
    {
      title: 'MongoDB',
      icon: ListIcon,
      description: 'A document-oriented database that stores data in JSON-like documents with flexible schema. MongoDB is known for its high performance, scalability, and ease of use. It is widely used in modern web and mobile applications for managing large amounts of unstructured data.',
      link: 'https://www.mongodb.com'
    }
  ]);
}
```

```
id: 1,
  title: 'React Components',
  description: 'Изучение функциональных и классовых компонентов',
  category: 'frontend',
  status: 'in-progress',
  createdAt: new Date('2024-01-15').toISOString()
},
{
  id: 2,
  title: 'Material-UI',
  description: 'Освоение Material Design для React',
  category: 'ui-library',
  status: 'not-started',
  createdAt: new Date('2024-01-10').toISOString()
},
{
  id: 3,
  title: 'React Hooks',
  description: 'Использование useState, useEffect и других хуков',
  category: 'frontend',
  status: 'completed',
  createdAt: new Date('2024-01-05').toISOString()
}
]);
};

const handleTabChange = (event, newValue) => {
  setTabValue(newValue);
};

const handleStatusChange = (techId, newStatus) => {
  setTechnologies(prev =>
    prev.map(tech =>
      tech.id === techId ? { ...tech, status: newStatus } : tech
    )
  );
};

const addNewTechnology = () => {
  const newTech = {
    id: Date.now(),
    title: `Новая технология ${technologies.length + 1}`,
    description: 'Описание новой технологии для изучения',
    category: 'other',
    status: 'not-started',
    createdAt: new Date().toISOString()
  };
  setTechnologies(prev => [...prev, newTech]);
};

return (
  <ThemeProvider theme={theme}>
    <CssBaseline />

    <AppBar position="static">
```

```
<Toolbar>
  <Typography variant="h6" component="div" sx={{ flexGrow: 1 }}>
    🔎 Трекер технологий
  </Typography>
</Toolbar>
</AppBar>

{/* Навигация табами */}
<Box sx={{ borderBottom: 1, borderColor: 'divider' }}>
  <Tabs value={tabValue} onChange={handleTabChange} aria-label="app tabs">
    <Tab icon={<ListItemIcon />} label="Список технологий" />
    <Tab icon={<DashboardIcon />} label="Дашборд" />
  </Tabs>
</Box>

{/* Вкладка списка технологий */}
<TabPanel value={tabValue} index={0}>
  <Container maxWidth="lg">
    <Box sx={{ textAlign: 'center', mb: 4 }}>
      <Typography variant="h4" component="h1" gutterBottom>
        Мои технологии
      </Typography>

      <Button
        variant="contained"
        startIcon={<Add />}
        onClick={addNewTechnology}
        size="large"
        sx={{ mb: 3 }}
      >
        Добавить технологию
      </Button>
    </Box>

    <Grid container spacing={3}>
      {technologies.map(technology => (
        <Grid item xs={12} sm={6} md={4} key={technology.id}>
          <SimpleTechCard
            technology={technology}
            onStatusChange={handleStatusChange}
          />
        </Grid>
      )));
    </Grid>
  </Container>
</TabPanel>

{/* Вкладка дашборда */}
<TabPanel value={tabValue} index={1}>
  <Dashboard technologies={technologies} />
</TabPanel>
</ThemeProvider>
);
}
```

```
export default App;
```

Практическая часть

Интеграция UI-кит в основное приложение

Шаг 1: Создайте Theme провайдер

```
// styles/theme.js
import { createTheme } from '@mui/material/styles';

export const theme = createTheme({
  palette: {
    primary: {
      main: '#1976d2',
      light: '#42a5f5',
      dark: '#1565c0',
    },
    secondary: {
      main: '#9c27b0',
    },
    background: {
      default: '#f5f5f5',
      paper: '#ffffff',
    },
  },
  typography: {
    fontFamily: '"Inter", "Roboto", "Helvetica", "Arial", sans-serif',
    h4: {
      fontWeight: 600,
    },
    h6: {
      fontWeight: 600,
    },
  },
  shape: {
    borderRadius: 8,
  },
  components: {
    MuiCard: {
      styleOverrides: {
        root: {
          boxShadow: '0 2px 8px rgba(0,0,0,0.1)',
        },
      },
    },
    MuiButton: {
      styleOverrides: {
        root: {
          textTransform: 'none',
        },
      },
    },
  },
});
```

```
        fontWeight: 600,
    },
},
},
},
}),
});

export const darkTheme = createTheme({
palette: {
mode: 'dark',
primary: {
main: '#90caf9',
},
secondary: {
main: '#ce93d8',
},
},
// ... остальные настройки
});
```

Шаг 2: Обновите главный App.js

```
// App.js
import React, { useState } from 'react';
import { ThemeProvider, CssBaseline, Box, Container } from '@mui/material';
import { theme } from './styles/theme';
import MuiDashboard from './components/MuiDashboard';
import MuiTechnologyCard from './components/MuiTechnologyCard';
import MuiTechnologyModal from './components/MuiTechnologyModal';
import { AppBar, Toolbar, Typography, Button, Grid } from '@mui/material';
import { Add as AddIcon } from '@mui/icons-material';

function App() {
const [technologies, setTechnologies] = useState([]);
const [isModalOpen, setIsModalOpen] = useState(false);
const [editingTech, setEditingTech] = useState(null);

const handleAddTechnology = (techData) => {
const newTech = {
id: Date.now(),
...techData,
status: 'not-started',
createdAt: new Date().toISOString()
};
setTechnologies(prev => [...prev, newTech]);
};

const handleEditTechnology = (techData) => {
setTechnologies(prev =>
prev.map(tech => tech.id === editingTech.id ? { ...tech, ...techData } : tech)
);
```

```
    setEditingTech(null);
};

const handleSaveTechnology = (techData) => {
  if (editingTech) {
    handleEditTechnology(techData);
  } else {
    handleAddTechnology(techData);
  }
};

const handleEdit = (technology) => {
  setEditingTech(technology);
  setIsModalOpen(true);
};

const handleDelete = (techId) => {
  setTechnologies(prev => prev.filter(tech => tech.id !== techId));
};

const handleStatusChange = (techId, newStatus) => {
  setTechnologies(prev =>
    prev.map(tech => tech.id === techId ? { ...tech, status: newStatus } : tech)
  );
};

return (
  <ThemeProvider theme={theme}>
    <CssBaseline />

    <Box sx={{ flexGrow: 1, minHeight: '100vh', backgroundColor:
      'background.default' }}>
      {/* Шапка приложения */}
      <AppBar position="static" elevation={2}>
        <Toolbar>
          <Typography variant="h6" component="div" sx={{ flexGrow: 1 }}>
            🚀 Трекер изучения технологий
          </Typography>
          <Button
            color="inherit"
            startIcon={<AddIcon />}
            onClick={() => {
              setEditingTech(null);
              setIsModalOpen(true);
            }}
          >
            Добавить технологию
          </Button>
        </Toolbar>
      </AppBar>

      <Container maxWidth="xl" sx={{ py: 3 }}>
        {/* Дашборд */}
        <MuiDashboard technologies={technologies} />
    
```

```
/* Сетка технологий */
<Box sx={{ mt: 4 }}>
  <Typography variant="h5" gutterBottom>
    Мои технологии ({technologies.length})
  </Typography>

  <Grid container spacing={3}>
    {technologies.map(technology => (
      <Grid item xs={12} sm={6} md={4} key={technology.id}>
        <MuiTechnologyCard
          technology={technology}
          onEdit={handleEdit}
          onDelete={handleDelete}
          onStatusChange={handleStatusChange}
        />
      </Grid>
    )))
  </Grid>

{technologies.length === 0 && (
  <Box
    textAlign="center"
    py={8}
    color="text.secondary"
  >
    <Typography variant="h6" gutterBottom>
      Технологий пока нет
    </Typography>
    <Typography variant="body1" gutterBottom>
      Добавьте первую технологию для отслеживания прогресса
    </Typography>
    <Button
      variant="contained"
      startIcon={<AddIcon />}
      onClick={() => setIsModalOpen(true)}
      sx={{ mt: 2 }}
    >
      Добавить технологию
    </Button>
  </Box>
)
}
</Box>
</Container>

/* Модальное окно */
<MuiTechnologyModal
  open={isModalOpen}
  onClose={() => {
    setIsModalOpen(false);
    setEditingTech(null);
  }}
  technology={editingTech}
  onSave={handleSaveTechnology}
>
```

```
    />
  </Box>
</ThemeProvider>
);
}

export default App;
```

Самостоятельная работа

Задание 1: Создайте компонент уведомлений с использованием Snackbar из MUI

Задание 2: Добавьте переключение темы (светлая/тёмная)

Что проверить перед завершением:

- Все компоненты MUI корректно отображаются
- Тема применяется ко всему приложению
- Модальные окна работают правильно
- Адаптивный дизайн работает на разных размерах экрана
- Иконки и интерактивные элементы понятны

Обратите внимание, что данная практика - последняя в блоке по React и последняя для выполнения соответствующей контрольной работы №4.