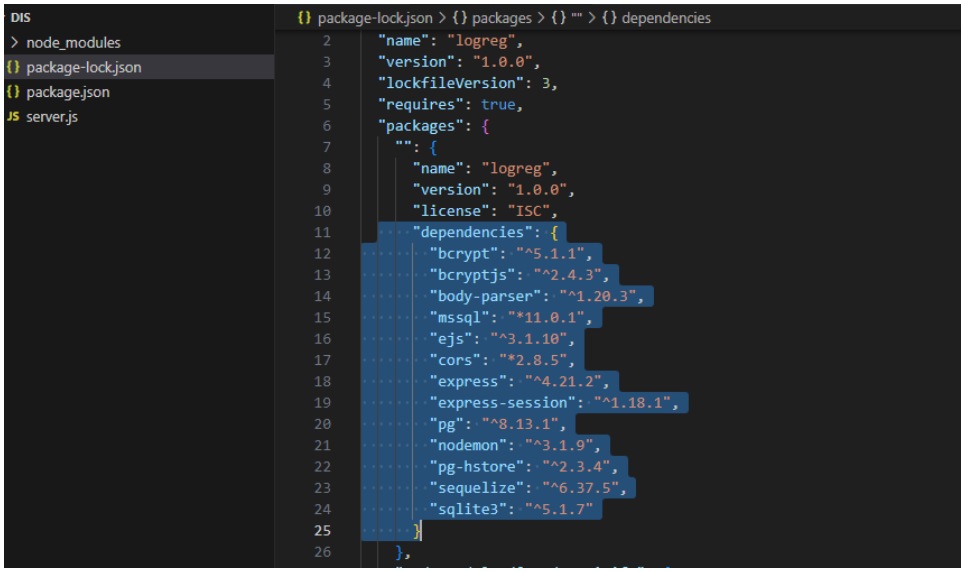
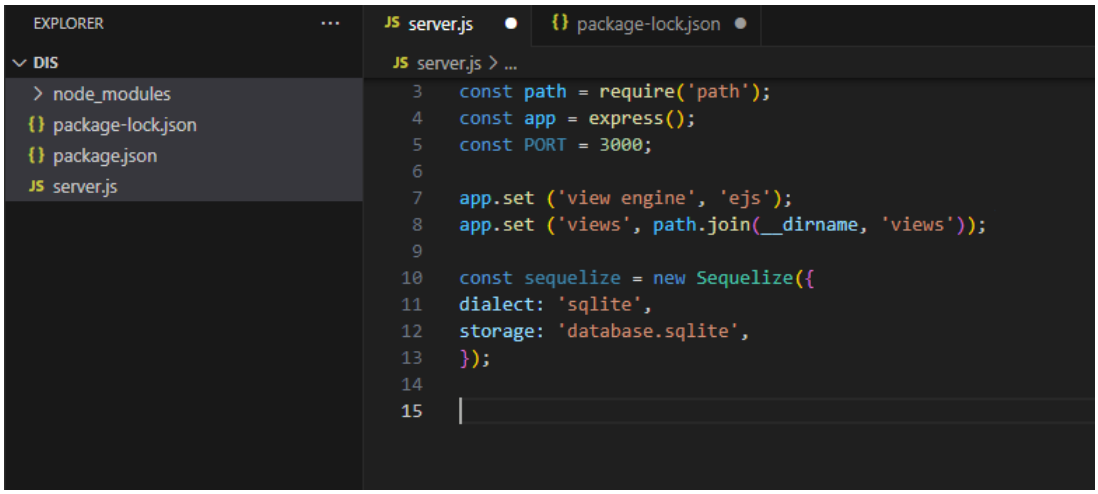
Отчет

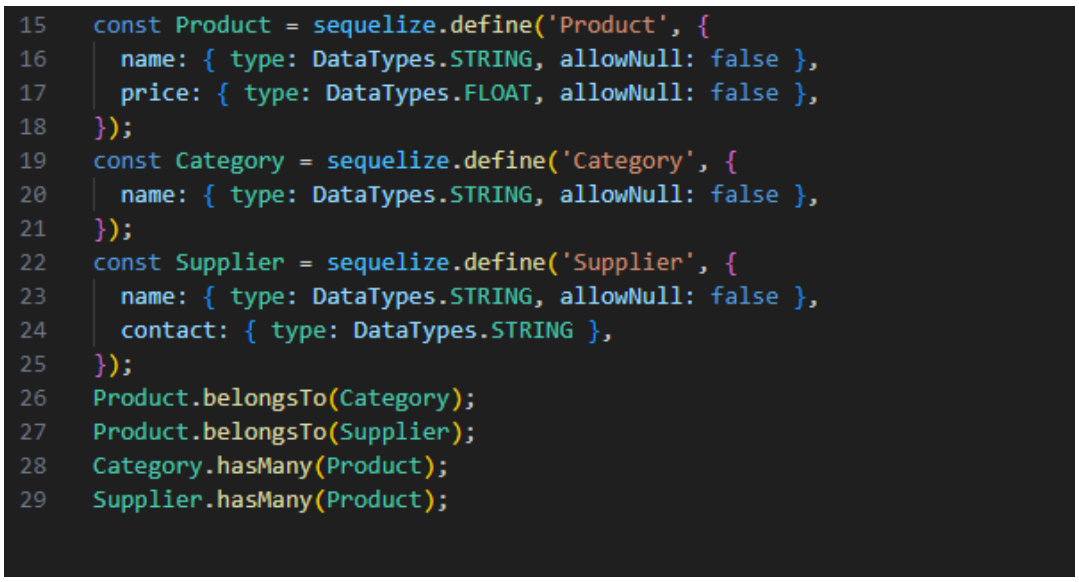
Инициализируем проект



Базовые настройки сервера



Создаем модель данных



Подключаем статические файлы и прописываем маршруты

```
app.get('/', async (req, res) => {
  const products = await Product.findAll({
    include: [Category, Supplier],
  });
  res.render('index', { products });
});
app.get('/add-category', (req, res) => {
  res.render('add-category');
});
app.get('/add-supplier', (req, res) => {
  res.render('add-supplier');
});
app.get('/add-product', async (req, res) => {
  try {
    const categories = await Category.findAll();
    const suppliers = await Supplier.findAll();
    res.render('add-product', { categories, suppliers });
  } catch (error) {
    console.error('Error fetching categories or suppliers:', error);
    res.status(580).send('Internal Server Error');
  }
});
```

```
app.get ('/edit-product/:id', async (req, res) => {
  try {
  const productId = req.params.id;
  const product = await Product.findByPk(productId, {
      include: [Category, Supplier],
  });
  const categories = await Category. findAll();
  const suppliers = await Supplier.findAl1();
  res. render ('edit-product', { product, categories, suppliers });
  } catch (error) {
  console.error('Error fetching product for edit:', error);
  res.status(580).send('Internal Server Error');
  }
});
```

Пост запросы

```
app.post('/add-category', express.urlencoded({ extended: true }), async (req, res) => {
  const { name } = req.body;
  if (name) {
      await Category.create({ name });
  }
  res.redirect('/');
  });
  app.post('/add-supplier', express.urlencoded({ extended: true }), async (req, res) => {
  const { name, contact } = req.body;
  if (name) {
      await Supplier.create({ name, contact });
  }
  res.redirect('/');
});

```

```
app.post('/add-product', express.urlencoded({ extended: true }), async (req, res) => {
  const { name, price, categoryId, supplierId } = req.body;
  if (name && price && categoryId && supplierid) {
    try {
  await Product.create({
  name,
  price,
  CategoryId: categoryId,
  SupplierId: supplierId,
});
  res. redirect('/');
 } catch (error) {
  console.error ('Error adding product:', error);
  res.status(500).send ('Internal Server Error');
 }
  } else {
  res.status(400).send('All fields are required');
  }
});
```

```
  app.post('/delete-product/:id', async (req, res) => {
  try {
      const productId = req.params.id;
      await Product. destroy({
        where: { id: productId },
  });
      res.redirect('/');
    } catch (error) {
      console.error('Error deleting product:', error);
      res.status(508).send('Internal Server Error');
}
});

  app.post('/edit-product/:id', express.urlencoded({ extended: true }), async (req, res) => {
    try {
      const productid = req.params.id;
      const { name, price, categoryId, supplierId } = req.body;

      await Product. update(
        { name, price, CategoryId: categoryId, SupplierId: supplierId },
        { where: { id: productid } }
      );

      res.redirect('/');
  } catch (error) {
      console.error( 'Error updating product:', error);
      res.status(580).send('Internal Server Error');
  }
  });
```

Начальный набор данный и подключение к серверу

```
(async () => {
    try {
        await sequelize.sync({ force: true });

  const category1 = await Category.create({ name: 'Electronics' });
  const category2 = await Category.create({ name: 'Books' });

  const supplier1 = await Supplier.create({ name: 'TechCorp', contact: 'techcorp@example.com' });
  const supplier2 = await Supplier.create({ name: 'BookStore', contact: 'contact@bookstore.com' });

  await Product.create({ name: 'Laptop', price: 1209.99, CategoryId: category1.id, SupplierId: supplieri.id });
  await Product.create({ name: 'Smartphone', price: 799.49, CategoryId: category1.id, SupplierId: supplier1.id });
  await Product.create({ name: 'Science Fiction Book', price: 19.99, CategoryId: category2.id, SupplierId: supplier2.id });

  app.listen (PORT, () => console. log(`Server is running on http://localhost:${PORT}`));
  } catch (error) {
  console.error ('Error initializing the application:', error);
  }
})();
```

Index.ejs

```
<body>
    <table>
        <thead>
            <th>Name</th>
            <th>Price</th>
            <th>Category</th>
            <th>Supplier</th>
            <th>Actions</th>
            </tr>
        </thead>
        <tbody>
            <%products.forEach (product=> { %>
                <tr>
                    <td><%= product.name %></td>
                    <td><%= product.price.toFixed(2) %></td>
                    <td><%= product.Category.name %></td>
                    <td><%= product.Supplier.name %></td>
                    <td>
                    <form action="/delete-product/<%= product.id %>" method="POST" style="display:inline;">
                        <button type="submit">Удалить</button>
                    </form>
                        <button onclick="window.location.href='/edit-product/<%= product.id %>'">Изменить</button>
                    </td>
                </tr>
                <% }) %>
        </tbody>
    </table>

</body>
```

```
    <div class="buttons">
        <button onclick="window. location.href='/add-product'">Добавить продукт </button>
        <button onclick="window. location.href='/add-category'">Добавить категориюк</button>
        <button onclick="window. location.href='/add-supplier'">Добавить поставщика</button>
</div>
```

Edit-product.ejs

```
<body>
    <h1>Изменить продукт</h1>
    <form action="/edit-product/<%= product.id %>" method="POST" >

    <label For="name">Название продукта: </label>
    <input type="text" id="name" name="name" value="<%= product.name %>" required>

    <label for="price">Цена:</label>
    <input type="number" id="price" name="price" value="<%= product.price %>" step="0.01" required>

    <label for="category">Выберите категорию:</label>
    <select id="category" name="categoryId" required>
<% categories.forEach(category => {%>
<option value="<%= category.id %>" <%= category.id === product.CategoryId ? 'selected' : '' %>><%= category.name %></optio
<% }) %>
</select>

<label for="supplier">Выберите поставщика:</label>
<select id="supplier" name="supplierId" required>
<% suppliers.forEach(supplier => { %>
<option value="<%= supplier.id %>" <%= supplier.id === product.SupplierId ? 'selected' : '' %>><%= supplier.name %></option
<% }) %>
</select>

<button type="submit">Сохранить</button>
</form>
```

Add-suplier.ejs

```ejs
<body>
    <h1>Добавить поставщика</h1>
    <form action="/add-supplier" method="POST">
        <label for="name">Название поставщика: </label>
        <input type="text" id="name" name="name" required>
        <label for="contact">онтактная информация: </label>
        <input type="text" id="contact" name="contact">
        <button type="submit">Сохранить</button>
    </form>
</body>

</html>
```

Add-category.ejs

```ejs
    <h1>Добавить категориюк</h1>
    <form action="/add-category" method="POST">
        <label for="name">Название категории:</label>
        <input type="text" id="name" name="name" required>
        <button type="submit">Сохранить</button>
    </form>
</body>
```

Add-product.ejs

```ejs
<body>
    <h1>Добавить продукт</h1>
    <form action="/add-product" method="POST">
        <label for="name">Название продукта:</label>
        <input type="text" id="name" name="name" required>

        <label for="price">Цена: </label>
        <input type="number" id="price" name="price" step="0.81" required>

        <label for="category">Выберите категорию:</label>
        <select id="category" name="categoryId" required>
        <% categories.forEach (category => { %>  c
            <option value="<%= category.id %>"><%= category.name %></option>
        <% }) %>
        </select>

        <label for="supplier">Выберите поставщика: </label>
        <select id="supplier" name="supplierId" required>
        <% suppliers.forEach(supplier => { %>
        <option value="<%= supplier.id %>"><%= supplier.name %></option>
        <% }) %>
        </select>

        <button type="submit">Сохранить</button>
    </form>
</body>
```

Результаты работы

SQL ▾ 1 / 1 1 - 2 of 2

id	name	contact	createdAt
1	TechCorp	techcorp@example.com	2024-12-16 21:29:30.123 +00:00
2	BookStore	contact@bookstore.com	2024-12-16 21:29:30.137 +00:00

SQL ▾ 1 / 1 1 - 3 of 3

id	name	price	createdAt	updatedAt	CategoryId	Suppl
1	Laptop	1209.99	2024-12-16 21:29:30.147 +00:00	2024-12-16 21:29:30.147 +00:00	1	1
2	Smartphone	799.49	2024-12-16 21:29:30.160 +00:00	2024-12-16 21:29:30.160 +00:00	2	1
3	Science Fiction Book	19.99	2024-12-16 21:29:30.172 +00:00	2024-12-16 21:29:30.172 +00:00	2	2

SQL ▾ 1 / 1 1 - 2 of 2

id	name	createdAt	updatedAt
1	Electronics	2024-12-16 21:29:30.082 +00:00	2024-12-16 21:29:30.082 +00:00
2	Books	2024-12-16 21:29:30.107 +00:00	2024-12-16 21:29:30.107 +00:00

Name	Price	Category	Supplier	Actions	
Laptop	1209.99	Electronics	TechCorp	Удалить	Изменить
Smartphone	799.49	Books	TechCorp	Удалить	Изменить
Science Fiction Book	19.99	Books	BookStore	Удалить	Изменить

Добавить продукт Добавить категориюк Добавить поставщика

Добавить продукт

Название продукта: Цена: Выберите категорию: >Electronics Выберите поставщика: TechCorp Сохранить

Добавить категориюк

Название категории: Сохранить

Добавить поставщика

Название поставщика: [ ] гонтактная информация: [ ] Сохранить