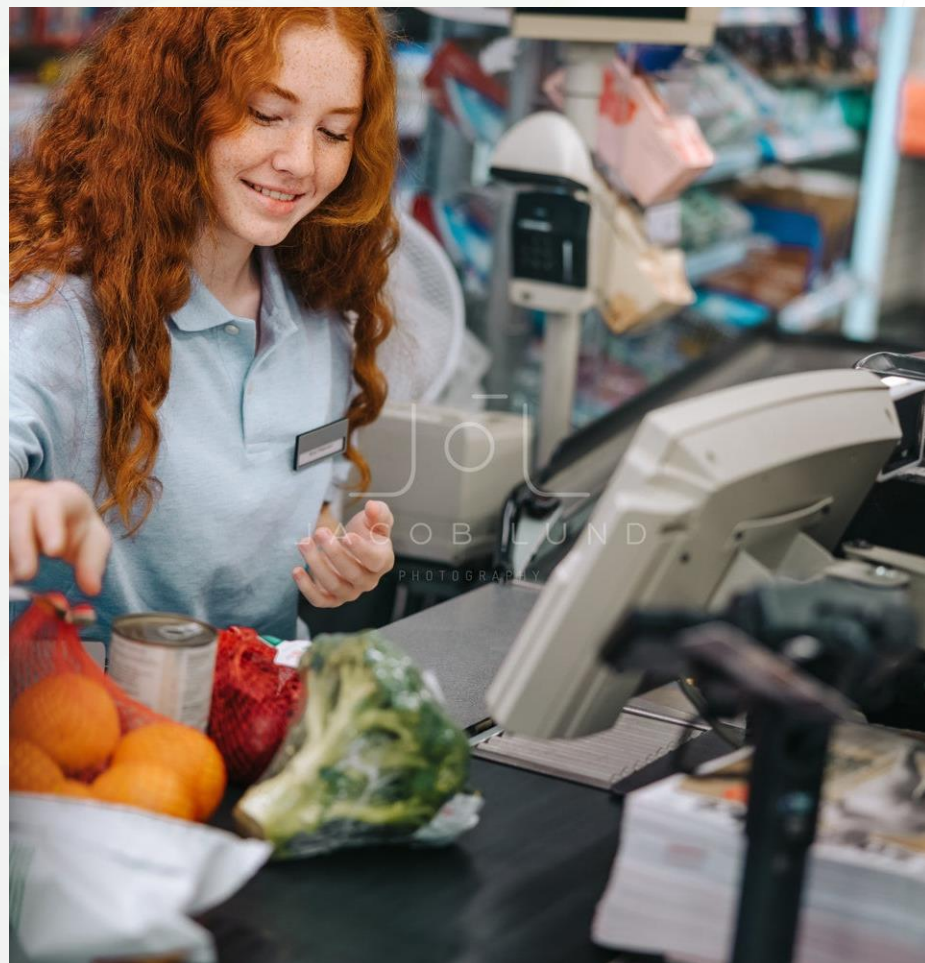




Store Register software



Developed by Vendel Hummel and Róbert Emánuel Hunka



The problem

- difficulty tracking products, supplier and employee data
- expensive software licenses

The solution

- affordable all-in-one software
- seamless integration
- additional features
- reduces everyday mistakes

Key features

Basic login system for both platforms

Full data manipulation for products

Functional cashier register software to manage transactions

Valuable insights to transactions and other data through charts

The program

Backend – Node.js with Express.js

- RESTful API endpoints
- Middleware for authentication
- Server-side data validation

Frontend – JavaScript, CSS

- Dynamic product cards
- Interactive forms for CRUD operations
- Desktop App - Windows Forms .NET Framework
- Cashier register software
- Create transactions and upload it to the database
- Export receipts

Database – MySQL

```
dotenv.config();
const connection = await mysql.createConnection({
  host: process.env.DB_HOST || 'localhost',
  port: process.env.DB_PORT || 3307,
  user: process.env.DB_USER || 'root',
  password: process.env.DB_PASSWORD,
  database: process.env.DB_NAME || 'db_nyilvantartas'
});
connection.connect((err) => {
  if (err) {
    console.error('Error connecting to the MySQL server: ' + err.stack);
    return;
  }
  console.log('Connected to the MySQL server.');
```

```
// Routers
import alkalmazott from './routes/alkalmazott.js';
app.use('/server/alkalmazott', alkalmazott);

import szamla from './routes/szamla.js';
app.use('/server/szamla', szamla);

import termek from './routes/termek.js';
app.use('/server/termek', termek);

import tetel from './routes/tetel.js';
app.use('/server/tetel', tetel);

import beszallito from './routes/beszallito.js';
app.use('/server/beszallito', beszallito);
```

Backend

Node.js with Express.js

- RESTful API endpoints
- Middleware for authentication
- Server-side data validation

Routers

- Router for each entity
- /server prefix
- Async functions
- Express.js, with middlewares

```
router.get('/payment-preference', async (req, res) => {  
  try {  
    const result = await WPaymentPreference();  
    res.json(result);  
  } catch (error) {  
    console.error('Error fetching payment preference:', error);  
    res.status(500).json({ error: 'Internal server error' });  
  }  
});
```

```
router.get('/:aazon', async (req, res) => {  
  try {  
    let alkalmazott = await DAlkalmazott(req.params.aazon);  
    res.status(201).json(alkalmazott);  
  } catch (error) {  
    console.error('DAlkalmazott error:', error);  
    res.status(500).json({ error: 'Internal server error' });  
  }  
});
```

localhost:3000/server/termek/all

Pretty-print ☐

```
[{"tazon":74,"tnev":"","tkategoria":"Pékáru","tar":0,"tmennyiseg":0,"tmennyisegiegység":"db","tminkeszlet":0,"trendeles":0,"tkoros":0,"bazon":0},  
{"tazon":75,"tnev":"","tkategoria":"Pékáru","tar":0,"tmennyiseg":0,"tmennyisegiegység":"db","tminkeszlet":0,"trendeles":0,"tkoros":0,"bazon":0},  
{"tazon":57,"tnev":"a","tkategoria":"Pékáru","tar":1,"tmennyiseg":2.1,"tmennyisegiegység":"kg","tminkeszlet":2,"trendeles":1,"tkoros":0,"bazon":6
```


Login router

```
router.get('/belepes/:felulet/:id', async (req, res) => {
  try {
    console.log(`Handling GET /server/alkalmazott/belepes/${req.params.felulet}/${req.params.id}`);
    const { felulet, id } = req.params;

    const numericId = Number(id);

    const result = await db.Belepes(felulet, numericId);

    if (!result) {
      console.log('Invalid felulet:', felulet);
      return res.status(400).json({
        success: false,
        message: 'Érvénytelen felület'
      });
    }

    if (result.length === 0) {
      console.log('No user found or no web access:', { felulet, id });
      return res.status(401).json({
        success: false,
        message: 'Érvénytelen felhasználói azonosító vagy nincs webes jogosultság'
      });
    }

    console.log('Login successful:', result);
    res.status(200).json(result);
  } catch (error) {
    console.error('Error in belepes:', error);
    res.status(500).json({
      success: false,
      message: 'Szerver hiba',
      error: error.message
    });
  }
});
```

Detailed Endpoints explanation

Endpoint	HTTP Method	Purpose	Router file
/server/termek/all	GET	Retrieves all products	termek.js
/server/termek	POST	Creates a new product	termek.js
/server/termek/:tazon	PUT	Updates a product by ID	termek.js
/server/termek/:tazon	DELETE	Deletes a product by ID	termek.js
server/szaml/:sazon	DELETE	Deletes transaction	szaml.js
/server/szaml/monthly-sales	GET	Gets the sum of sales each month from the past 12 months	szaml.js

Frontend

Frontend – JavaScript, CSS

- Modern CSS with responsiveness in mind
- Interactive forms for CRUD operations
- Dynamic product cards
- Login system

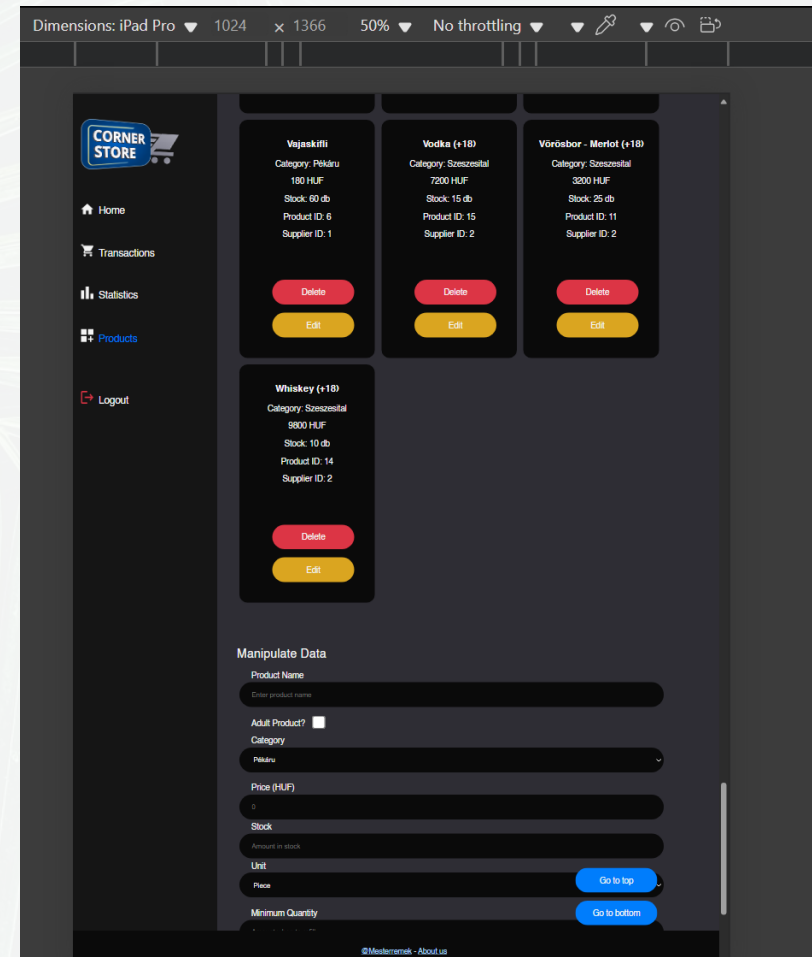
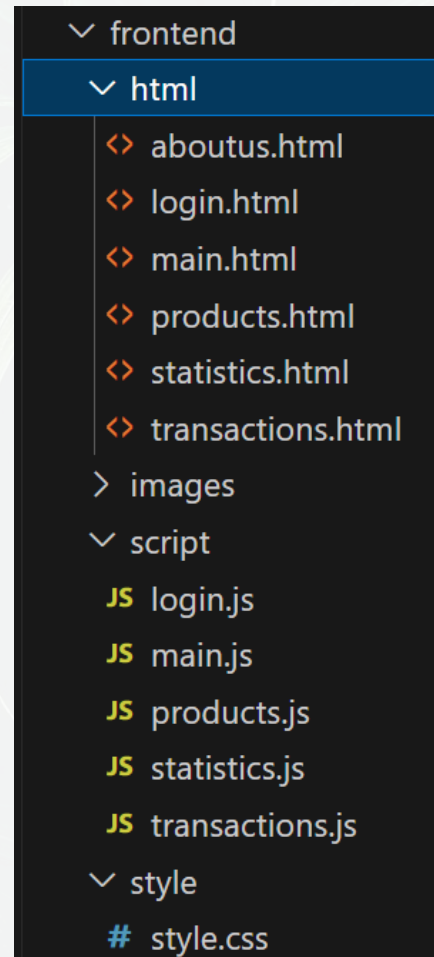
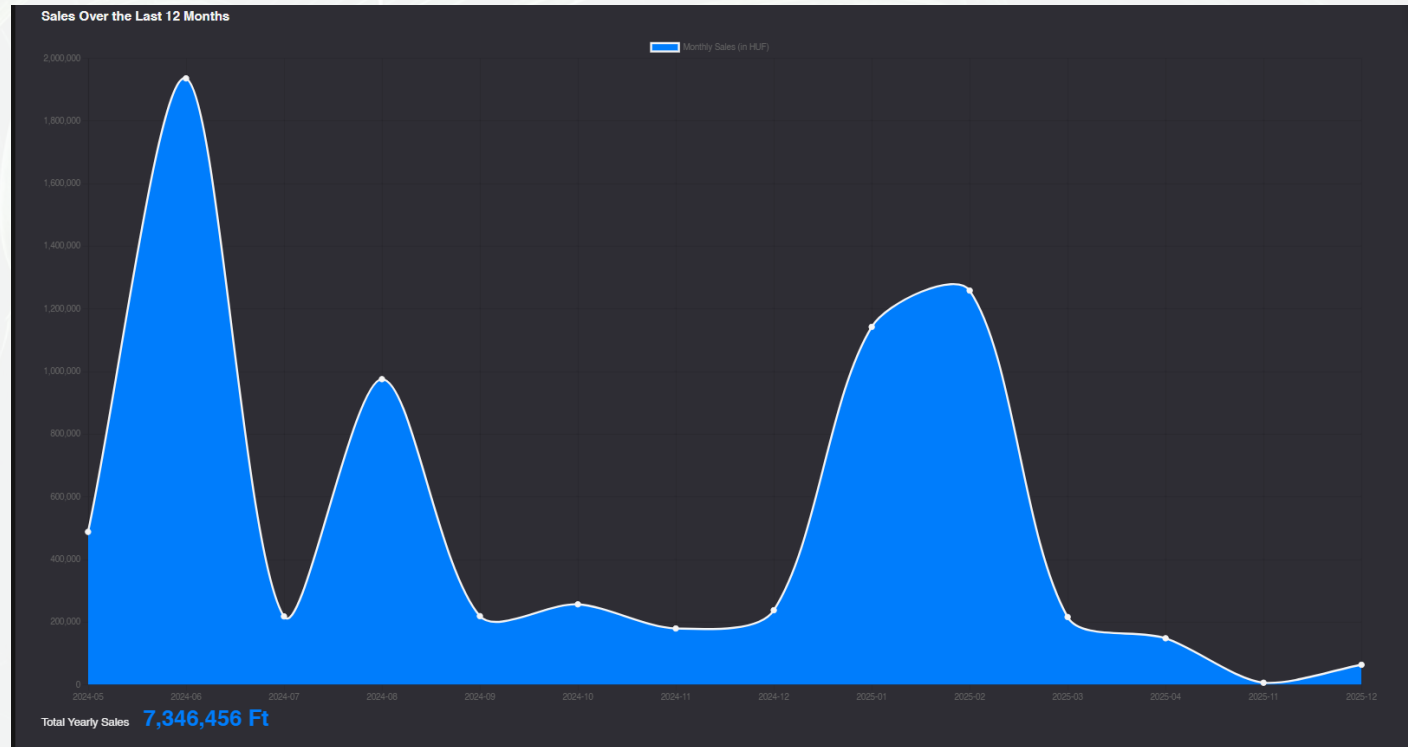


Chart.JS API

Clean and
responsive
statistics




```
.then(data => {  
  sum_yearlysales = Number(data.reduce((sum, item) => sum + item.total_sales, 0)).toLocaleString();  
  monthlySales.insertAdjacentHTML('afterend', `Total Yearly Sales <span style="color: var(--signaturecolor); font-weight: bold; padding: 1rem; font-size:2rem"> ${sum_yearlysales} Ft<span></p>`);  
  new Chart(monthlySales, {  
    type: 'line',  
    data: {  
      labels: data.map(item => item.month),  
      datasets: [{  
        label: 'Monthly Sales (in HUF)',  
        data: data.map(item => item.total_sales),  
        fill: true,  
        borderColor: '■ rgba(245, 245, 245, 1)',  
        backgroundColor: '■ rgba(0, 125, 252, 1)',  
        tension: 0.3,  
        pointBackgroundColor: '■ rgba(245, 245, 245, 1)',  
        pointRadius: 4  
      }]  
    },  
    options: {  
      responsive: true,  
      plugins: {  
        legend: {  
          position: 'top'  
        }  
      },  
      scales: {  
        y: {  
          beginAtZero: false  
        }  
      }  
    }  
  })  
})
```


Desktop

Windows Forms .NET Framework

- Record transactions from costumers
- User-friendly design
- Login system

CORNER STORE Pénztár 6

Termék azonosító:

Kategória: Termék:

Egységár:

Mennyiség:

Összeg:

Tétel hozzáadása

0 Ft

Utolsó tétel törlése

Szia Nagy Péter! Jó újra látni.

OK

Fizetési mód: ☐ Kártya ☒ Készpénz

Fizetendő összeg:

Fizetett összeg:

Számla rögzítése

Termék azonosító:

37

Kategória:

Gyümölcs

Termék:

Banán

Egységár:

600 Ft / kg

Mennyiség:

6,80

Összeg:

4080 Ft

Tétel hozzáadása

Termék azonosító:

2

Kategória:

Pékáru

Termék:

Bagett

Egységár:

350 Ft / db

Mennyiség:

3

Összeg:

1050 Ft

Tétel hozzáadása

Termék azonosító:

14

Kategória:

Szeszesita

Termék:

Whiskey

Figyelem, a termék korosztályhoz kötött!

Egységár:

9800 Ft / db

Mennyiség:

1

Összeg:

9800 Ft

A tételhez megerősítés szükséges!

Tétel hozzáadása

Pénztár

CORNER STORE

Pénztár 6

Termék azonosító: 14

Kategória: Szeszesita Termék: Whiskey

Egységár: 9800 Ft / db

Mennyiség: 1

Összeg: 9800 Ft

A tételhez megerősítés szükséges! ☒

Tétel hozzáadása

Banán	2 kg * 600 Ft/kg	Összesen: 1200 Ft
Whiskey	1 db * 9800 Ft/db	Összesen: 9800 Ft

Utolsó tétel törlése

Végösszeg:

11000 Ft

Fizetési mód: Kártya • Készpénz ○

Fizetendő összeg: 11000 Ft

Fizetett összeg: 20000 Ft

Visszajáró: 9000 Ft

A kosár tartalma jóváhagyást igényel! ☒

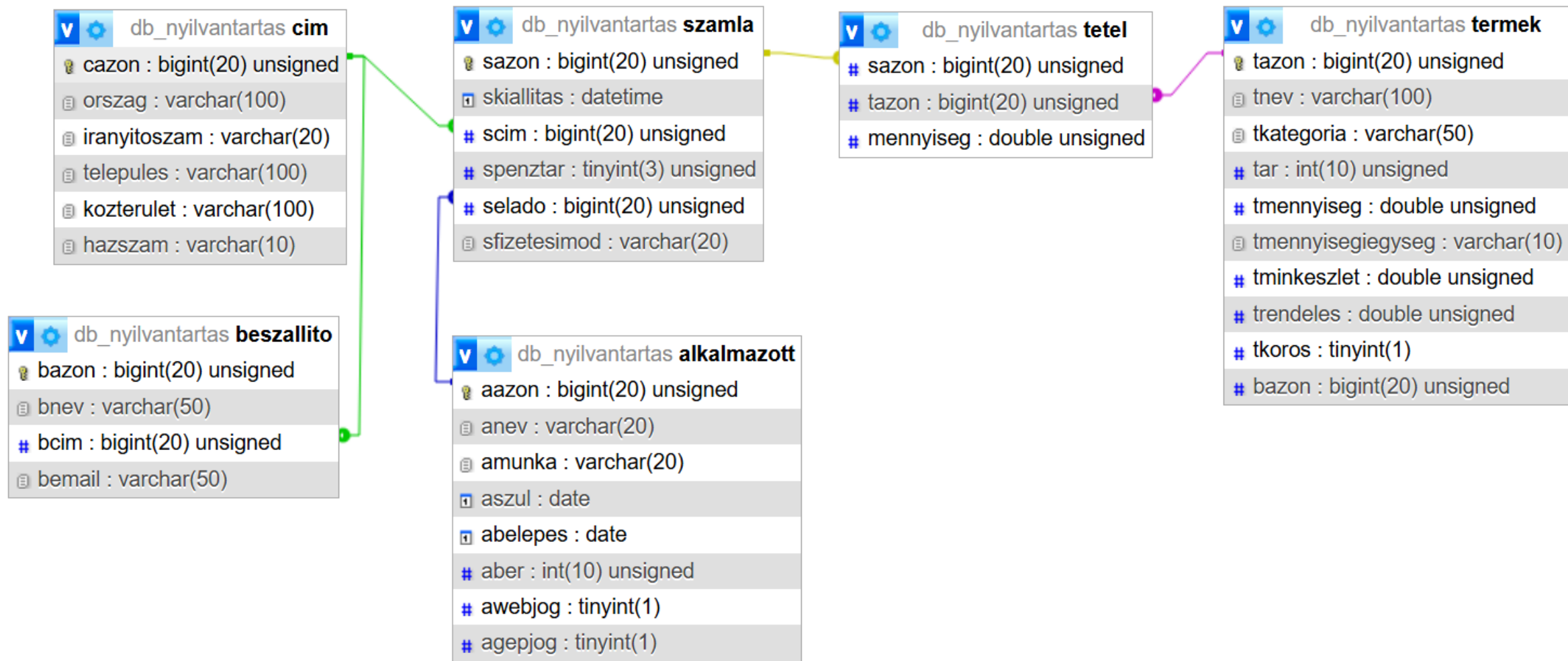
Számla rögzítése

- Divided to panels
- Great readability
- Consistent UI

Database

- MySQL based
- Includes:
 - o Employees, products, transactions, etc.
- 6 tables
- 5 foreign keys





db_nyilvantartas.sql

Tests performed

API Endpoint Tests

- Tested all the endpoints to ensure it retrieves all data correctly, checking for proper data formatting and response status (200 OK).

Frontend Functionality Tests

- Tested the web interface's product management by adding, editing, and deleting products through the form, ensuring the UI updates and notifications (e.g., "Product updated successfully") appear correctly.

Forms testing

- Tested form submissions to ensure data is correctly formatted and compatible with the MySQL database (e.g., proper data types, required fields).

Performance and Usability Tests

- Tested the system with a large dataset (e.g., 100+ items from receipts) to ensure the web interface and API responses remain fast and responsive.
- Conducted usability tests on both platforms, ensuring the UI is intuitive for adding products, processing sales, and viewing data insights simultaneously

General Teamwork

- The core idea
- Assigning tasks
- Backend Endpoints
- Documentation



Our tools:

- GitHub (publication + version control)
- Discord and Microsoft Teams
- Figma
- Trello



CORNER STORE

Pénztár 1

Termék azonosító:

Kategória: Termék:

Egységár: 2 000 Ft / Kg

Mennyiség:

Összeg: 6 000 Ft

A tételhez megerősítés szükséges! ☒

Tétel hozzáadása

3 db	Magos zsemle	350 Ft / db	1050 Ft
3 db	Magos zsemle	350 Ft / db	1050 Ft
3 db	Magos zsemle	350 Ft / db	1050 Ft
3 db	Magos zsemle	350 Ft / db	1050 Ft
3 db	Magos zsemle	350 Ft / db	1050 Ft
3 db	Magos zsemle	350 Ft / db	1050 Ft
3 db	Magos zsemle	350 Ft / db	1050 Ft
3 db	Magos zsemle	350 Ft / db	1050 Ft
3 db	Magos zsemle	350 Ft / db	1050 Ft
3 db	Magos zsemle	350 Ft / db	1050 Ft

Fizetési mód: ☐ Kártya ☒ Készpénz

Fizetendő összeg: Fizetett összeg:

Végösszeg: 11 550 Ft

Utolsó tétel törlése

Store register ☆ 👤 📄 Tábla ▼ 🔍 ⚡ ⚙ Szűrők 🗑️ 🔄 V.H. ➡ Megosztás ...

General Information

- Description and conditions 2
- Requirements and rules 2
- Communication tools

+ Kártya hozzáadása

Brainstorm

- Figure out what the project will be about
- Find a name for our project 1
- Product's structure and tools to use
- Assign tasks

+ Kártya hozzáadása

Version Control

- Setup GitHub repository
- Create README.md 0/6
- Create codeDocumentation.md
- Create technicalDocumentation.md

+ Kártya hozzáadása

Database

- Plan out the sturcture for the database
- Create the empty database
- Fill up database with data

+ Kártya hozzáadása

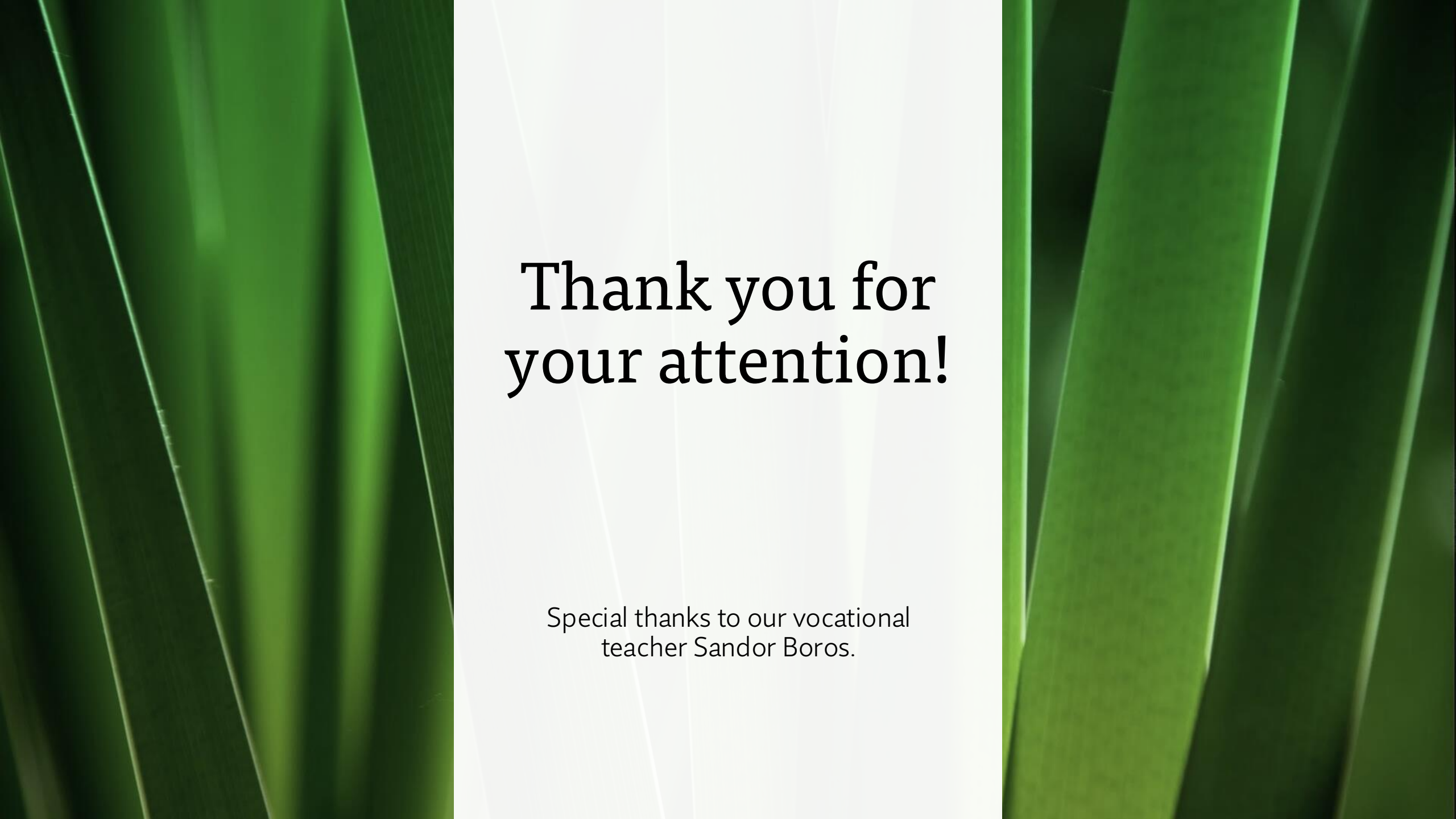
Server Endpoint

- Write the serve
- Implement end server

+ Kártya hozz

The future

1. Export statistics
2. Admin panel
3. Complete transactions data manipulation
4. Multi-language support for a broader audience

The slide features a central white rectangular area with text. On the left and right sides of this white area are vertical borders composed of several overlapping, slightly blurred green leaf-like shapes, creating a natural, organic frame.

Thank you for your attention!

Special thanks to our vocational
teacher Sandor Boros.