Имена: Али Надер, Дамян Георгиев, Никола Топалов
фн: *5MI0600080, 4MI0600145, 1MI0600248*
Начална година: 2025 **Програма:** бакалавър, СИ **Курс:** 3
Тема: **54.1 Система за поздравителни картички - единични или по групи**
Дата: **2025-06-04** **Предмет:** **w21ed-001_SI_final** **имейл:**
damiangeorgiev03@gmail.com, 18alinader18@gmail.com, nikolatopalov10@gmail.com

преподавател: доц. д-р Милен Петров

ТЕМА: w21ed-001/54.1/

Система за поздравителни картички - единични или по групи

1. Условие

Целта на проекта е да се подготвят персонализирани масово картички за изпращане (и/или споделяне) до множество получатели - такъв пример може да бъде изпращане на 'картичка' (html/pdf/tex страница и файл) - изпраща , като конфигурира основните параметри на картичката от csv файл - по един ред на получател. Например контакти с бизнес партньори, картички до студенти послучай 8ми декември и т.н.; Създаване на картички и персонализиране; Генериране на картичките и изпращане/споделяне - съгласно различен набор от опции

2. Въведение – извличане на изисквания

Роли:

- администратор на системата - Настройва сървъра, потребителите и интеграциите (SMTP, API ключове).
- потребител - Импортира CSV, избира шаблон, конфигурира изпращането (единично, групово).
- получател - Получава персонализирана картичка по имейл или линк, има опция за споделяне.

Функционални изисквания:

- Регистрация и вход на потребители.
- Импортиране на CSV с персонализирани данни (имена, имейли, съобщения, позиция/размер на стикери и фон).
- Визуално редактиране на картичка в реално време.
- Добавяне и преместване на стикери, избор на фонове, шрифт и размер на текста.
- Изпращане на картички чрез SMTP (или записване в локална папка в демо режим).
- Запазване на текуща картичка като CSV за бъдеща редакция.
- Галерия със стикери и шаблони.
- Преглед на изпратени картички (в режим на разработка).

Нефункционални изисквания:

- Съвместимост с локална среда (без нужда от постоянен интернет).

- Сигурно съхранение на данни и опростено потребителско изживяване.
- Лесно разширяване на набора от фонове и стикери.

Теория – анализ и проектиране на решението

Системата е разделена на няколко модула:

3.1 Интерфейс (Frontend)

- Реализиран чрез HTML, CSS и JavaScript.
- Основните страници са: `homepage.html`, `create-card.html`, `login.php`, `register.php`.
- JS логика се съдържа във `card-builder.js`, който управлява взаимодействието между формите, визуализацията на картичката и зареждането на изображения и стикери.

3.2 Бекенд (Backend)

- Реализиран чрез Node.js (за изпращане на имейли чрез `server.js`) и PHP (за регистрация и вход).
- `process_login.php` и `process_register.php` използват MySQL за валидиране и съхранение на данни.
- Сървърната част за имейли използва `nodemailer` с поддръжка на `.env` файл за конфигурации.

3.3 База данни

- Използва се MySQL база данни с таблица `users` (потребителско име, парола, имейл).
- Потребителите се валидират чрез сесии, задавани от PHP.

3.4 Обработка на CSV

- Зареждане на CSV файл с персонализирани данни.
- Генериране на картички на базата на шаблон с динамично добавяне на стикери, фон, съобщение
- Съхраняване на позицията и размера на стикери, както и избрания текст.

4. Използвани технологии

Софтуер и езици:

- HTML5, CSS3 – за структура и стил на потребителския интерфейс.
- JavaScript (ES6) – динамична логика, визуализация на картички.
- PHP 7+ – регистрация, логин, управление на сесии.
- Node.js 18+ – изпращане на имейли чрез `nodemailer`.
- MySQL 5.7+ – база данни за потребителски акаунти.
- CSV – импортиране/експортиране на данни за картички.

Библиотеки и модули:

- `nodemailer` – за изпращане на прикачени картички по имейл.
- `dotenv` – за конфигурации през `.env` файл.
- `express`, `cors` – за сървъра и API endpoint.

Операционна система и среди:

- Локална среда: Windows 10, Visual Studio Code, GitHub Desktop.
- Сървърна част: Node.js локален сървър, XAMPP (за PHP и phpMyAdmin).

Пътища и адреси:

- Стартова страница: `http://127.0.0.1:5500/homepage.html`
- Достъп до база данни: `http://localhost/phpmyadmin`
- SMTP чрез Gmail или демо режим (локално записване).

5. Инсталация, настройки и DevOps

5.1 Необходими инсталации

За да стартирате проекта успешно на локален компютър, е необходимо да бъдат инсталирани:

- **XAMPP** – за поддръжка на PHP и MySQL (https://www.apachefriends.org/index.html)
- **Node.js** – за работа с бекенд скрипта за изпращане на имейли (https://nodejs.org/)
- **VS Code / текстов редактор** – за редакция на код и настройки

5.2 Структура на проекта

Папка / файл	Съдържание
`/htdocs/login/`	Файлове, свързани с логин/регистрация (PHP + база)
`/htdocs/card/`	HTML + JS за създаване на картички
`/server/`	Node.js скрипт за изпращане на PNG файлове по имейл
`/img/`	Фонове и стикери за картичките

5.3 Настройки на база данни (MySQL)

1. Стартирайте **XAMPP**, включете **Apache** и **MySQL**
2. Отворете в браузър:
   http://localhost/phpmyadmin
3. импортирай базата данни с име `card_users_table.sql`

4. Проверете дали в `process_login.php` и `process_register.php` настройките за свързване са:

```
php
```

```
$host = "localhost";

$user = "root";

$password = "";

$dbname = "card_users";
```

5.4 Стартиране на проекта

PHP част (регистрация и вход):

- Сложете `login.php`, `register.php`, `process_*.php` в `C:/xampp/htdocs/login/`
- Стартирайте с:
  `http://localhost/login/login.php`

Node.js част (имейл изпращане):

1. Отидете в папка `/server`

Създайте `.env` файл с данни:

```ini
GMAIL_USER=вашият_имейл@gmail.com

GMAIL_PASS=парола_или_app_password
```

2. Инсталирайте зависимости:

```nginx
npm install
```

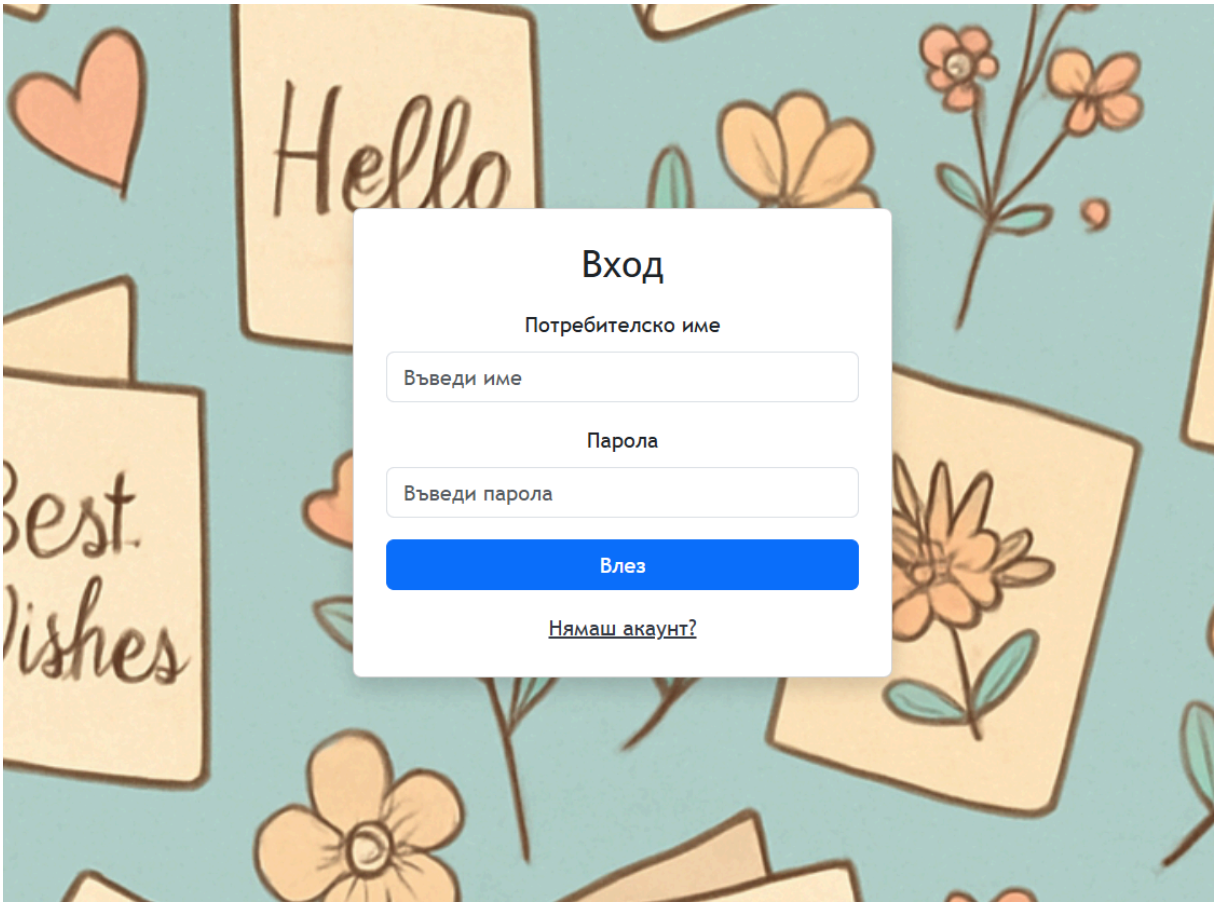
3. Стартирайте сървъра:

```nginx
node server.js
```

6. Кратко ръководство на потребителя

6.1. Вход

Проста имплементация за Регистрация и Вход.



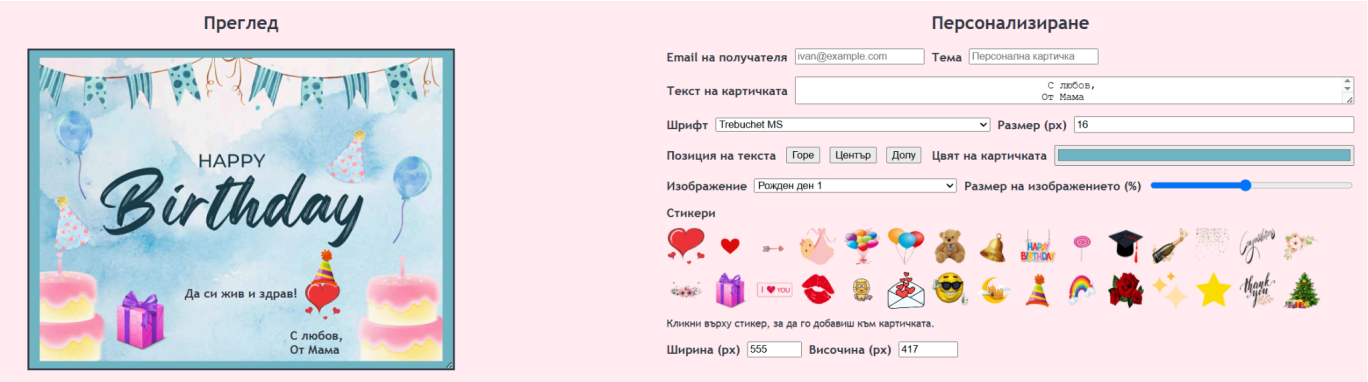
фигура 1. Посреща ви екрана за Вход

6.2. Създаване на картичка

След отваряне на страницата “Създай картичка”, се вижда предварителен изглед на картичката, разположен в центъра на екрана. Потребителят може да персонализира следните елементи:

- Текст на картичката – въведете желаното съобщение в полето "Съобщение".
- Шрифт – изберете от падащото меню предпочитан стил на шрифта.
- Размер на шрифта – регулира се чрез плъзгач или въвеждане на стойност в поле.
- Цвят на фона – с color picker може да изберете желания фон на картичката.
- Размер на картичката – чрез полета за "ширина" и "височина" може да се определи точният размер в пиксели. Той автоматично се отразява в предварителния изглед. Също може да се променя чрез дърпане в долния десен ъгъл.
- Позиция на текста – Горе, Център, Долу.
- Фоново изображение - изберете изображение от падащото меню с картинки, които автоматично ще се заредят като фон.
- Стикери – с мишката можете да добавите стикери към картичката, които после могат да бъдат:

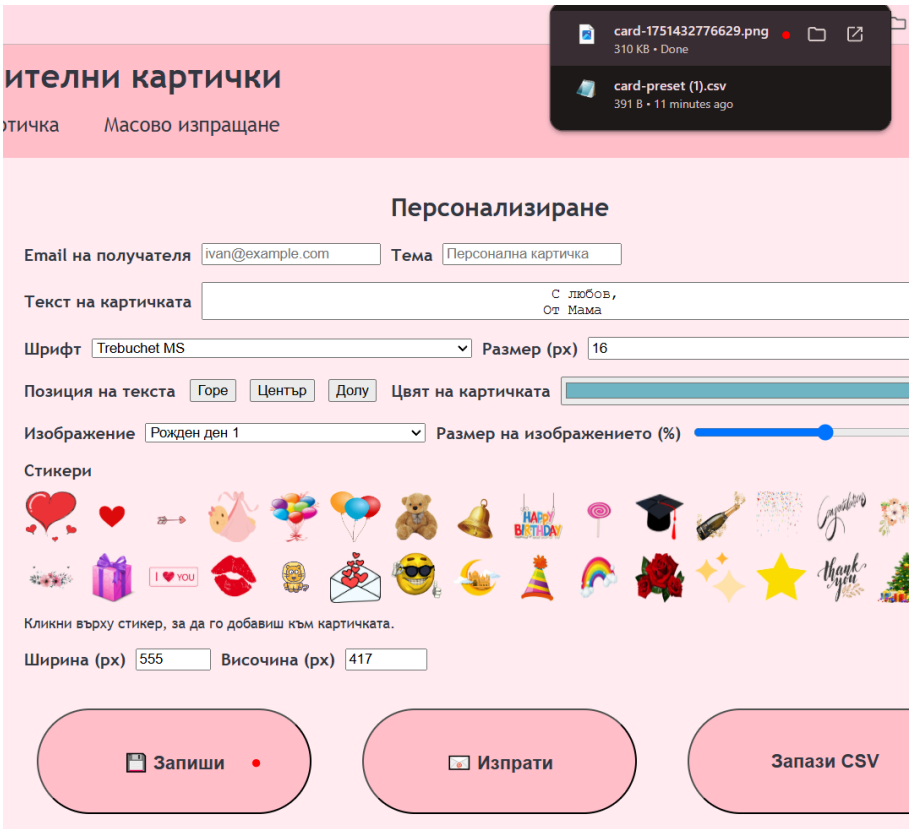
- местени (драгване),
- оразмерявани с плъзгане на ъгъл,
- изтривани чрез натискане на бутона Delete, когато са маркирани.



фигура 1. Примерна картичка за рожден ден

6.3. Запазване като PNG

Кликнете бутона **„Запиши“**, за да съхраните текущата картичка като `.png` файл на вашия компютър. Картичката ще бъде точно това, което виждате на екрана – с текст, цветове, фон и стикери.

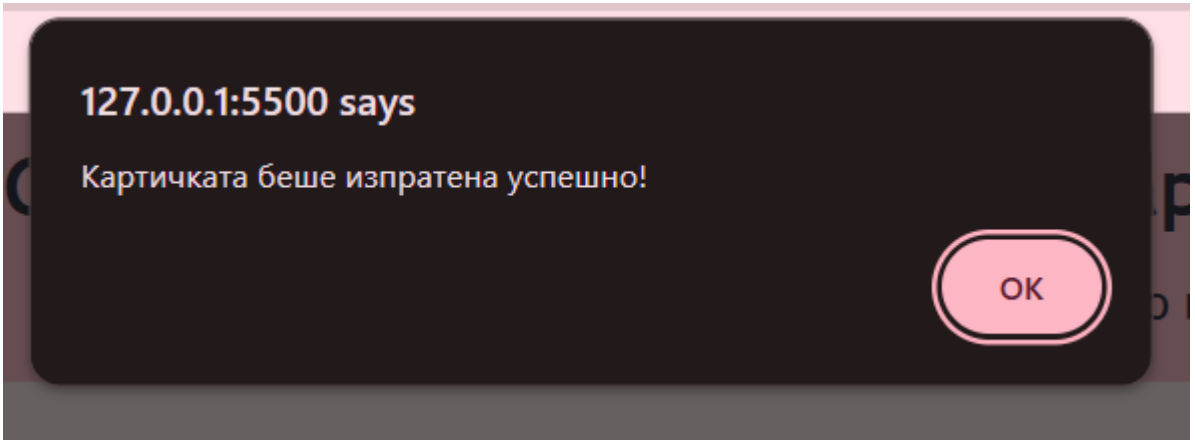


фигура 2. След натискане на бутона картичката се запазва

фигура 3. Резултат

6.4. Изпращане на един получател по Имейл

1. Въведете имейл адрес в полето „Email на получателя“.
2. При желание – редактирайте полето за тема („Subject“).
3. Натиснете бутона **„Изпрати“**.
4. Ще получите съобщение за успешно изпращане или грешка, ако нещо не е попълнено.



фигура 4. Успешно изпращане

6.5. Запазване като CSV

Ако сте създали картичка ръчно и искате да я използвате като шаблон за други хора, можете да натиснете бутона **„Запази като CSV“**, което ще създаде ред с текущите данни – шрифт, размер, фон, стикери, позиция и др. Този ред може после да бъде копиран или модифициран и ползван за бъдещо масово изпращане.

```
email,subject,message,font,fontSize,bgColor,cardW,cardH,image,imgSize,pos,stickers
"","","Да си жив и здрав!


                С любов,
                От Мама","Trebuchet
MS","16","#70b6c7","560","422","img/deco-birthday.png","95","85%","stk-heart.png@364,306,46
;stk-gift.png@108,316,80;stk-partyhat.png@363,260,54"
```

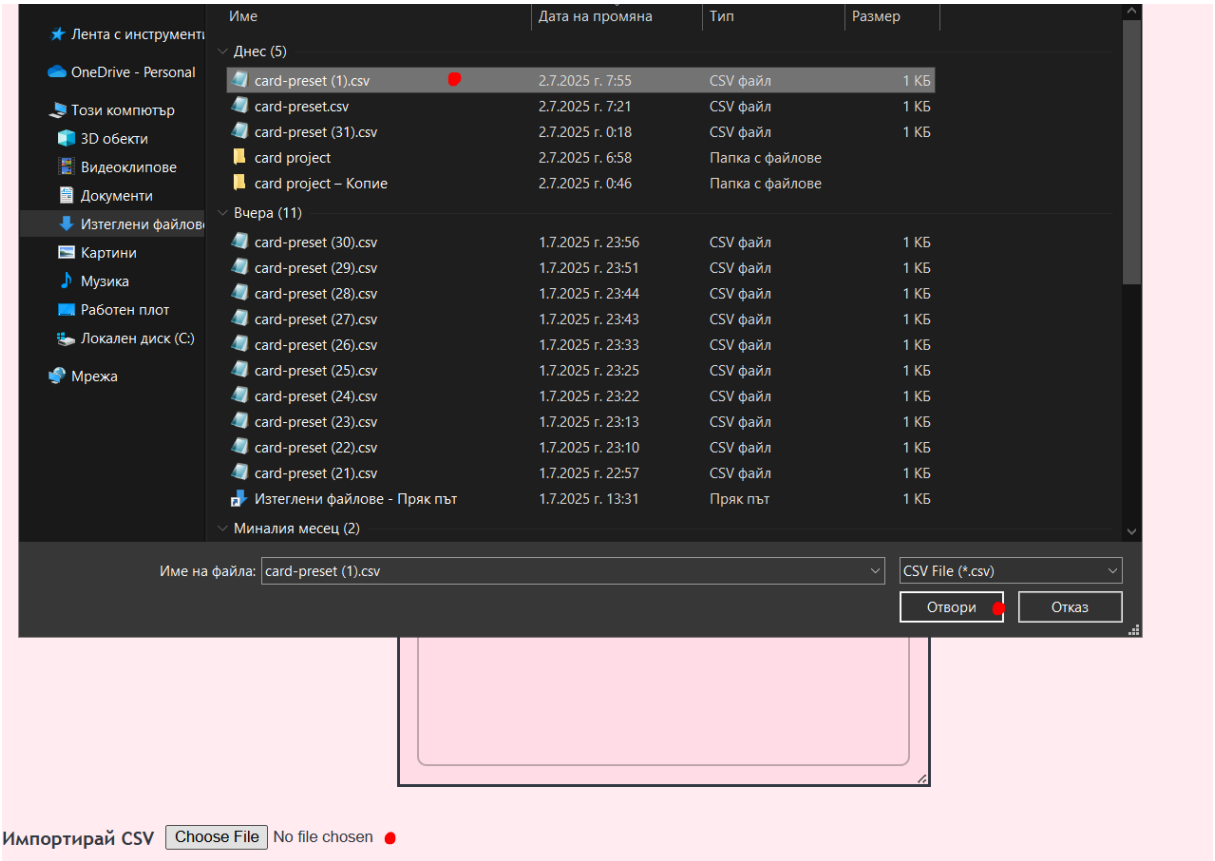
фигура 4. Съдържание на запазения .csv файл след натискане на бутона

6.6. Масово изпращане

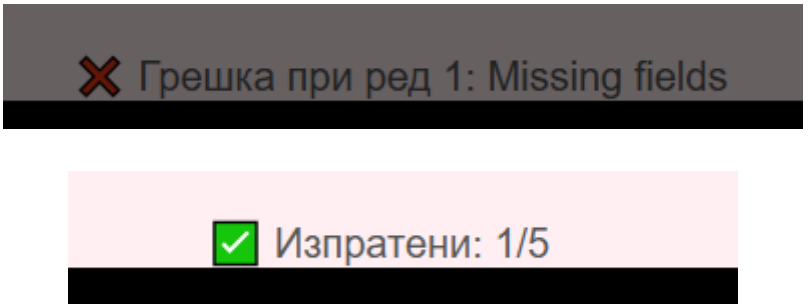
Приложението поддържа масово създаване и изпращане на картички по данни от `.csv` файл.

Стъпки:

1. Подгответе CSV файл (или използвайте генерирания от 6.5) с колони като:
   - `email`, `subject`, `message`, `font`, `fontSize`, `bgColor`, `cardW`, `cardH`, `image`, `imgSize`, `stickers`, `pos`
2. Заредете го чрез бутона **"Импортирай CSV"**.
3. Програмата автоматично ще генерира и изпрати всяка картичка на съответния получател, показвайки прогрес на броя изпратени (`Изпратени: 1/5`).

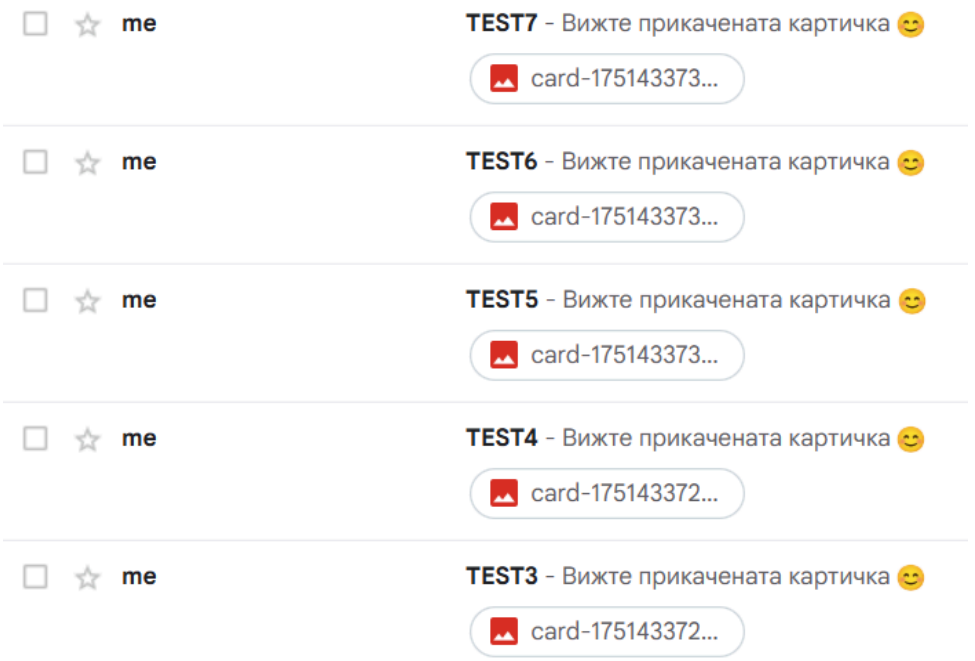


фигура 5. Избира се .csv файл от компютъра



фигура 6. При липса на имейл дава грешка/ следи прогреса

фигура 7. Получени картички



в имейл

7. Примерни данни

За тестване на системата са подготвени няколко примерни акаунта, CSV файлове и изображения, които демонстрират възможностите на уеб приложението. Всички необходими файлове са включени в структурата на проекта и са разположени в съответните папки, описани по-долу:

Примерен акаунт (за вход)

Тестването на логин системата може да се извърши с един от следните акаунти:

Потребителско име	Парола

AliNader test1

Забележка: За да функционира входът, проектът трябва да бъде стартиран в `htdocs/` чрез XAMPP или друг локален уеб сървър с активна MySQL база.

Разположение на тестови файлове и изображения

Папка / Файл	Съдържание и предназначение
`example.csv`	Съдържа примерен CSV файл за масово създаване и изпращане на картички
`mass_sentexample.png`	Съдържа PNG файлове, симулиращи изпратени картички (при тест в offline/demo режим)
`/img/`	Папка с фонови изображения и стикери за картичките
`create-card.html`	Главната страница за създаване и редакция на картичка
`homepage.html`	Начална страница след успешен вход

🧪 Тестови CSV файл – (`example.csv`)

csv

```
to,message,image,textFont,textSize,textColor,sticker1X,sticker1Y,sticker1Size
```

```
test@example.com,"Честит рожден ден,
Ани!","img/deco-flowers.png","Georgia",24,"#000000",120,150,85
```

```
demo@example.com,"Весели празници,
Петър!","img/deco-love.png","Trebuchet MS",20,"#FF0000",200,100,60
```

Действия за тестване

1. Стартирайте локалния сървър (Apache + MySQL).
2. Отворете `http://localhost/login.php`, въведете един от тестовите акаунти.
3. След вход – пренасочване към `homepage.html`.
4. Натиснете „Създай картичка“ → зарежда се `create-card.html`.
5. Оттам:
    - Тествайте ръчно създаване и редакция.
    - Импортирайте CSV от `/test_csv/sample.csv`.
    - Натиснете бутона „Изпрати“ или „Запази PNG“.

Допълнителна информация

- Всички тестови изображения са във формат `.png` или `.svg`.
- Шрифтовете са стандартни и не изискват интернет връзка.
- Ако не работи изпращането по имейл, уверете се, че Node.js сървърът е стартиран чрез `node server.js`.

8. Описание на програмния код

Проектът представлява **уебсайт за създаване и изпращане на персонализирани поздравителни картички**, който включва **регистрация, вход, визуален редактор на картички, импортиране от CSV и изпращане по e-mail**. Проектът е структуриран в няколко основни модула и файла, всеки със своя роля:

1. **Формата за login и register:**

config.php: Това е **конфигурационен файл**, който установява връзка с MySQL базата данни. Използва се обект `mysqli` за създаване на връзката. Ако връзката е неуспешна, скриптът спира с `die()` и показва грешка.

process_login.php: Зарежда `config.php` и стартира сесия. Взима въведените потребителско име и парола от формата (`POST` заявка). Изпълнява подготвена SQL заявка, за да извлече потребителя от базата по потребителско име. Ако потребителят съществува:

- Сравнява въведената парола с хешираната от базата чрез `password_verify()`;
- При успех се съхранява информация за потребителя в сесията.

При неуспех – извежда съобщения за грешка.

process_register.php: Получава потребителските данни от `POST` формата. Хешира паролата със `password_hash()` преди да я запише в базата. Проверява дали вече има потребител със същото име или имейл. Ако не – записва новия потребител в базата. При успех – извежда съобщение и линк към `login.php`.

```
// Фрагмент 1 – Пренасочване след успешен вход

$_SESSION['username'] = $user['username'];

header("Location: ../homepage.html");
```

create_users_table.sql: Създава база данни `user_auth`, ако не съществува.След това създава таблица `users` с колони:

- `id`: уникален идентификатор, автоматично нарастващ;
- `username`, `email`: уникални и задължителни;
- `password`: съхранява хешираната парола;
- `created_at`: автоматично записва времето на регистрация.

validate.js: Функцията validateLogin(): Взимане на стойностите:

- С `document.querySelector('[name="username"]')` се избира полето по име.
- `value.trim()` премахва излишните интервали от началото и края.

2. **Проверка за празни полета**:
   - Ако някое от двете полета (`username` или `password`) е празно, се показва съобщение.
   - `return false;` спира изпращането на формата.
3. **Успешна валидация**:
   - Ако всичко е попълнено, функцията връща `true` → формулярът се изпраща.

Използва се в `login.php`, чрез атрибут `onsubmit="return validateLogin()"`.

Функцията validateRegister():

1. **Събиране на въведените стойности**:
   - Извличат се всички полета от формуляра за регистрация.
2. **Проверка за празни полета**:
   - Ако някое от полетата е празно → съобщение и прекратяване.
3. **Сравняване на пароли**:
   - Ако паролата и потвърждението не съвпадат → съобщение.
4. **Минимална дължина на паролата**:
   - Ако паролата е под 6 символа → не се позволява регистрация.
5. **Валидна регистрация**:

- Всички условия са изпълнени → `true` → формата може да се изпрати.

Използва се в `register.php`, чрез атрибут `onsubmit="return validateRegister()"`.

logout.php: Изчиства сесията, прекратява достъпа на потребителя и пренасочва го към `login.php`.

2. `homepage.html` – Начална страница на сайта

Съдържа навигация към създаване на нова картичка, галерия и друга информация. В зависимост от сесията, показва бутон „Вход“ или „Изход“.

3. `create-card.html` – Интерфейс за създаване на картички

Тук потребителят избира фоново изображение, въвежда съобщение, избира шрифт, позиционира текста и добавя стикери. Включва и бутони за запис в PNG и за изпращане по имейл.

4. `card-builder.js` – Основна клиентска логика

Този JavaScript файл съдържа всички функционалности за динамично редактиране на картичките, включително визуални стилове, обработка на импортиран CSV, добавяне на стикери и др.

Ключови фрагменти от `card-builder.js`:

js

```
// Фрагмент 2 – Обновяване на съобщението при писане
msgInput.addEventListener('input', e => {
  txt.textContent = e.target.value || ' ';
});


// Фрагмент 3 – Добавяне на стикер и възможност за влачене
const stk = document.createElement('img');
stk.src = src;
stk.className = 'sticker-el';
stk.style.left = '50%'; stk.style.top = '50%';
```

```
preview.appendChild(stk);
makeDraggable(stk);
```

js

```
// Фрагмент 4 – Изпращане на PNG картичка чрез Node.js бекенд
const res = await fetch('http://localhost:3000/send-card', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({ to, subject, fileName, mime, data })
});
```

js

```
// Фрагмент 5 – Импортиране от CSV с PapaParse
Papa.parse(file, {
  complete: async res => {
    for (let row of res.data){
      msgInput.value = row.message;
      ... // Попълване на UI, добавяне на стикери, изпращане
    }
  }
});
```

5. `styles.css` – Основен стил на интерфейса

Файлът определя оформление на бутоните, картичката, стикерите, формулярите и изгледа на галерията. Поддържа responsive дизайн, съвместим с различни размери на екрана.

9. Приноси на студента, ограничения и възможности за бъдещо разширение

Никола Топалов: Формата за login и register.

Али Надер: Персонализация на картички и изпращане по имейл

Дамян Георгиев: Масово изпращане чрез CSV

10. Какво научих (най-важните неща, които сте научили по време на курса и при разработването на проекта-за всеки студент)

Никола Топалов:

Какво всъщност научих? По време на разработката на този проект имах възможност да приложа на практика знанията, придобити в рамките на курса, както и да науча нови неща, свързани с реалното изграждане на динамични уеб приложения.

На първо място, се запознах по-задълбочено с изграждането на система за регистрация и вход на потребители, която включва управление на сесии, работа със сървърна логика на PHP, както и комуникация с база данни чрез MySQL.

Разбрах как да структурирам уеб проект с ясно разделение между логика (PHP), оформление (CSS) и клиентска функционалност (JavaScript). В проекта използвах JavaScript за валидация на формулярите от страна на клиента, така че грешките да се засичат още преди изпращането на данни към сървъра.

Работата с HTML и CSS ми помогна да усъвършенствам уменията си за изграждане на прости и интуитивни потребителски интерфейси. Усвоих прилагането на стилове, подравняване, цветове, фонове и оформяне на бутони и формуляри.

В по-широк план, курсът ми даде стабилна основа в: Създаване и стилизиране на уеб страници с HTML5 и CSS3, основи на работа с PHP и връзка с бази данни и организация и структуриране на файлове и уеб приложения.

Разработката на проекта затвърди тези знания, като ми даде реален практически опит. Научих колко е важно да се мисли както за функционалността, така и за сигурността и удобството на крайния потребител.

Али Надер:

По време на курса и разработването на проекта научих как да създавам реално уеб приложение с помощта на HTML, CSS, JavaScript и Node.js. Разбрах как работи връзката между фронтенд и бекенд, как се изпращат данни чрез CSV и как да използвам по-добре DOM манипулация. Също така усвоих основите на работа с git и GitHub и как се структурират модули и функционалности в по-голям проект. Най-важното – придобих увереност, че мога да завърша цялостна уеб система самостоятелно.

Дамян Георгиев:

При разработването на проекта научих как се реализира масово изпращане на персонализирани съобщения чрез CSV файл. Запознах се с форматирането на данни, зареждането им в уеб форма и свързването с функционалности за генериране и изпращане на картички. Усвоих практически как да използвам CSV за динамично управление на съдържание и как се валидират данни при масов импорт. Научих също и как да структурирам проекта така, че да е лесен за използване от други.

11. Използвани източници

MDN Web Docs
Заглавие: *HTML, CSS и JavaScript документация*
Линк: https://developer.mozilla.org/
Посетено на: 01.07.2025
Автор: Mozilla Contributors

W3Schools
Заглавие: *HTML/CSS/JS/Node.js синтаксис и примери*
Линк: https://www.w3schools.com/

Stack Overflow
Заглавие: *Решения при технически проблеми с DOM, CORS и Node.js*
Линк: https://stackoverflow.com/
Посетено в периода: 25.06 – 02.07.2025

ChatGPT – OpenAI
Заглавие: *Интерактивна помощ при писане на код, обработка на грешки и структуриране на проекта*
Линк: https://chat.openai.com/
Посетено многократно между: 24.06 – 02.07.2025

Предал (подпис): …….

/*5MI0600080, 4MI0600145, 1MI0600248,* Али Надер, Дамян Георгиев, Никола Топалов, СИ/

Приел (подпис): …….

/проф. д-р *Милен Петров*/