A logo with yellow and blue text

AI-generated content may be incorrect.

Дипломен проект   
на  
Тема: „Проектиране и изграждане на келнер система“

Професия: Системен програмист

Специалност: Системно програмиране

Изготвил: Мартин Галинов Михалев

Ръководител: инж. Димитър Аврамов

[1. Увод 4](#_Toc196685079)

[2. Реализация 4](#_Toc196685080)

[2.1. Хардуерна реализация 4](#_Toc196685081)

[2.1.1. Fusion360 6](#_Toc196685082)

[2.1.2. EasyEDA 7](#_Toc196685083)

[2.2. Софтуерна реализация 8](#_Toc196685084)

[2.2.1. Фирмуер 8](#_Toc196685085)

[2.2.1.1. Функции в кода 8](#_Toc196685086)

[2.3.2. АПИ 17](#_Toc196685087)

[2.3.2.2. Visual Studio Code 20](#_Toc196685088)

[2.3.2.3. Node JS 21](#_Toc196685089)

[2.3.2.4. Express.js 22](#_Toc196685090)

[2.3.3. Сървър 23](#_Toc196685091)

[2.3.3.1. XAMPP 24](#_Toc196685092)

[2.3.3.2 phpMyAdmin 25](#_Toc196685093)

[2.3.3.3. MySql 26](#_Toc196685094)

[2.3.4. Визуалното приложение 27](#_Toc196685095)

[2.3.4.2. Visual Studio Community 2022 30](#_Toc196685096)

[2.3.4.3. C# Forms 31](#_Toc196685097)

[3. Инсталация и поддръжка 32](#_Toc196685098)

[4. Бъдеща реализация 33](#_Toc196685099)

[5. Заключение 34](#_Toc196685100)

[6. Използвана литература 34](#_Toc196685101)

1. Увод

Целта на келнер системата е да забърза и подобри процеса на поръчване в ресторанти и заведения за бързо хранене, като елиминира често срещани проблеми като забравена поръчка и дългото чакане на сервитьор за получаване на поръчката. Системата осигурява модерно решение на тези проблеми като съчетава в себе си хардуер и софтуер за да гарантира желанието на клиента.

В съвременната ресторантьорска индустрия бързината и ефективността на обслужването играят ключова роля за удовлетвореността на клиентите. С оглед на това, келнер системата е разработена за интегрирано управление на поръчки в заведения за хранене, която улеснява работата на сервитьорите, подобрява комуникацията с кухнята и оптимизира процеса на обслужване.

Келнер системата комбинира хардуерни и софтуерни компоненти в една цялостна платформа, като използва ESP32 микроконтролер, дисплей тип ILI9341, база данни, хоствана през XAMPP и phpMyAdmin, както и софтуерен потребителски панел, разработен на C# във Visual Studio Community. За сървърната логика се използва Node.js, а при нужда от 3D моделиране и корпуси се прилага Fusion 360.

Целта на системата е да намали времето за приемане и предаване на поръчки, да елиминира необходимостта келнер да си записва или да помни поръчките, както и да предостави ясна визуализация на поръчките към кухнята. Клиентът може лесно да въведе поръчка чрез графичен интерфейс, която се изпраща към сървъра, съхранява се в база от данни и се показва на кухненския дисплей.

1. Реализация
   1. Хардуерна реализация

Хардуерната част на келнер системата представлява основата за физическото взаимодействие между потребителя и софтуерната част на системата. За целта се използват ESP32, 2.1.2, дисплей ILI9341 ,които работят заедно, за да осигурят стабилна и интуитивна работа. На схемата е показано свързването им. ESP32 е подходящ за проекта заради неговате вградене Wi-Fi и Bluetoothспособности, което го прави изключително подходящ за IoT приложения. Той комуникира със сървъра чрез HTTP заявки, управлява дисплея и обработва получените от него данни. Микроконтролерът осигурява голям набор от входно-изходни пинове,двуядрен процесор с честота до 240MHz, поддръжка на SPI портокол за комуникация нужен за дисплея и позволява програмиране с Arduino IDE. За визуализация на менюто на клиента се използва 2.8 инчов TFT дисплей с резолюция 320x240 пиксела и контролер ILI9341. Той е избран за този проект заради неговата големина и вградена функция за допир за интуитивно управление. Компонентите са ръчно запоени върху перфорирана платка, съгласно свързването, показано на електрическата схема(Фигура 1), проектирана в EasyEDA. Този метод позволява стабилно и трайно свързване на елементите, като същевременно запазва гъвкавост при прототипиране и корекции по време на разработката. На диплея се визуализира продукти и категории, както и бутони за навигация. Корпусът който съхранява комплекта на батерията, микроконтролера, и диплея е 3D принтиран с PET-G A diagram of a circuit board

AI-generated content may be incorrect.материял и е проектиран в Fusion360.

Фигура 1- Електрическа схема на системата

A black rectangular electronic device with white text

AI-generated content may be incorrect.

Фигура 2- Снимка на системата

* + 1. Fusion360

Като платформа за 3D проектиране и инженерна разработка е използван Fusion 360. Предназначена е за създаване и симулация на технически и индустриални проекти в широк спектър от инженерни дисциплини. Създадена от Autodesk, тази мощна CAD/CAM/CAE среда предоставя богат набор от инструменти, които обединяват проектиране, инженеринг, симулации и производство в една интегрирана платформа. С Fusion 360 инженерите и дизайнерите могат да работят с параметрично и свободно 3D моделиране, повърхности, сглобки и чертежи. Платформата поддържа функционалности за симулация на натоварване, движение и термични анализи, както и инструменти за фрезоване, рязане и 3D печат. Тя е подходяща както за индивидуални създатели и студенти, така и за инженери в индустриална среда. Fusion 360 е известна със своето облачно базирано сътрудничество, което позволява на екипите да работят съвместно по проекти в реално време. Предлага версия контрол, управление на файлове и възможност за преглед и коментиране на модели директно в браузър. Това улеснява комуникацията и съкращава времето за проектиране. Освен това Fusion 360 поддържа разширения и допълнителни модули, които разширяват функциите на платформата – от генеративен дизайн до интеграция с CAM постпроцесори. Без значение дали става въпрос за механични компоненти, корпуси, прототипи или цялостни машини, Fusion 360 предоставя надежден и иновативен инструментариум за съвременно инженерно проектиране и производство.



Фигура 3- Лого на Fusio360

* + 1. EasyEDA

За проектирането на електронни схеми и печатни платки (PCB) в проекта е използван EasyEDA – онлайн платформа за електронно проектиране, която комбинира удобен интерфейс с мощни инструменти за създаване на схеми, симулации и PCB разпечатки. EasyEDA позволява бързо проектиране на електронни компоненти, проследяване на връзки и създаване на реалистични печатни платки.Платформата е достъпна директно през браузър, без нужда от инсталация, и предлага богата библиотека от готови компоненти, както и възможност за създаване на собствени. EasyEDA поддържа както симулация на електрически схеми, така и автоматично или ръчно разполагане на елементи върху платката, което я прави подходящ избор за бързо и ефективно прототипиране.

Фигура 4- Лого на EasyEDA

* 1. Софтуерна реализация

Софтуерната част на Келнер системата е отговорна за логиката на обслужване, съхранението на данни . Основните компоненти включват REST API който работи с помощта на MySql команди, сървър съхраняващ база от данни и потребителски интерфейс. Тук сущо се намира и кодът отговорен за микроконтролерът.

* + 1. Фирмуер

Кодът за микроконтролера (фирмуерът) е написа в среда за разработка Arduino IDE

* + - 1. Функции в кода
* Header

A screenshot of a computer program

AI-generated content may be incorrect.Тук се дефинират нужните библиотеки.Те включват SPI.h, TFT\_eSPI.h и XPT2046\_Touchscreen.h за визуализирането и получаване на информация от диспея, и WiFi.h, HTTPClient.h и ArduinoJson.h се осъществява връзката с wifi и с АПИ.

Фигура 5- Header

* Глобални променливи, обекти и структури

Тук се дефинират константи за името и паролата на безжичната мрежа и базовият линк за АПИ с IP на машината хост след това се декларират нужните обекти за дисплея. Дефинира се големината на дисплея,текста и нужните му пиновете . Създават се структури за категория, включваща Id и име, структура за ястие, включваща ид, име, цена, тегло и масиви от низове за съставки и алергени, A screen shot of a computer program

AI-generated content may be incorrect.дефинират се променливи за категория, артикул, етап и начислена сума.

Фигура 6- Глобални променливи

* Функция drawRowText()

A computer screen shot of a program code

AI-generated content may be incorrect.Използвасе за да може ако текстът е твърде дълъг, той да се прехвърли на другия ред.

Фигура 7- Функцията drawRowText()

* Функция initUI()

Тази функция се използва за визуализация на бутоните.

A computer screen shot of a program

AI-generated content may be incorrect.

Фигура 8-Функцията initUI()

* Функция setup()

Това е функция която винаги се изпълнява в началото на включването на системата. Вътре първо се осъществява връзка с безжичната мрежа, след което се инициализират обектите на дисплея определя се ориентацията на диплея и се извиква функцията initUI() за интерфейса.

A computer screen shot of a program

AI-generated content may be incorrect.

Фигура 9- Функцията setup()

* 1. Функция fetchCategory()

Тази функция е отговорна за получаването и визуализирането на категориите. В нея се получват категориите от АПИ-то като сериализирана информация след което с помощта на ArduinoJson тя се десериализира и се предава на Category cats[] (масив от структури).След като е получена информацията бива визуализиран първата категория в масива начислената за момента сума чрез променливата total и след това се прави проверка дали клиента е натиснал дисплея и дали допирната точка отговаря на някой от бутоните. Ако допирната точка отбоваря на бутона за надолу или нагоре съответно се премена изобразената категория, но ако допирната точка отговаря на бутона за напред, категорията се запазва в глобалната променлива selectedCategory и се преминава към следващия етап чрез stage=1.

A computer screen shot of a program code

AI-generated content may be incorrect.

Фигура 10- Първа част на функцията fetchCategory

A screen shot of a computer program

AI-generated content may be incorrect.

Фигура 11- Втора част на функцията fetchCategory

* Функция fetchByCategory(int category)

Това са използваните функции за получаването и визуализирането на продукт по дадени ID на тяхната категория. Тук се получват продуктите от категорията като сериализирана информация от тип JSON ,изпратена от АПИ-то, която трябва да бъде десериализирана и придадена на масив от продукти, като и в този момент биват взети алергените и съставките чрез функцииите fetchAllergies и fetchIngredients. След като продуктите се правилно запазени, на първия продукт бива изписани: името, теглото, цената, категорията,съставките и алергените.След това се прави проверка за допир от страна на клиента и ако клиентът е натиснал бутоните за нагоре или надолу се изобразява следващият или предишният продукт, но ако е натиснал бутона за напред избраният продукт бива запазен в глобалната променлива selectedDish и stage приема стойност 2 ,за да се премине към следващияетап, но ако клиентът е натиснал бутона за назад тови бива пратен A screen shot of a computer program

AI-generated content may be incorrect.A computer screen shot of a program

AI-generated content may be incorrect.къмминалия етап чрез stage=0.

Фигура 12- Първа част на функцията fetchByCategory

Фигура 13- Втора част на функцията fetchByCategory

* Функциите fetchAllergies(Dish& dish) и fetchIngredients(Dish& dish)

Тези функции са отговорни за получването на алергените и съставките на един продукт. Те иамт почти тенакъв синтакс и преимат параметър за показтел към продукт, като получват от АПИ-то сериализиран масив от имената на алергенитеи съставките за дадения продукта по ID-то му.

Фигура 14- Функцията fetchIngredients

* Функцията SendOrder()

Тази функция е отговорна за изпращането на поръчката. Първо се прави проверка дали това е желания продукт, като си изобразява нотификация за продуктът който ще бъде изпратен и ако клиента натисне бутона за назад поръчката ще бъде отказана и клиента ще бъде върнат в етапа за избиране на категория, ако обаче потребителят натисне бутона за напред, тогава бива микроконтролерът подготвя данни за изпращане във JSON формат. Създава се обект от тип DynamicJsonDocument съдържащ в себе си ID на маста и ID на желания продукт, след това той се сериализира и се изпраща чрез HTTP POST заявка. Цената на продукта бива добавена към total и се връща на първия етап чрез stage=0.

A screen shot of a computer program

AI-generated content may be incorrect.A computer screen shot of a program

AI-generated content may be incorrect.

Фигура 15- Втора част на функцията SendOrder

Фигура 16- Първа част на функцията SendOrder

* Функцията loop()

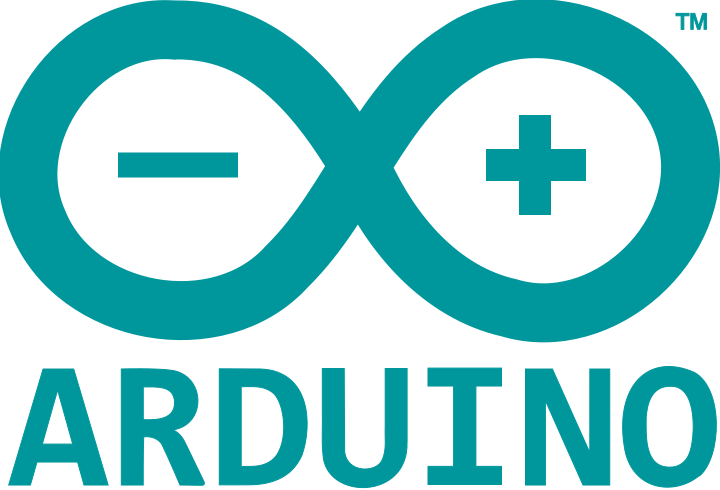
A screen shot of a computer

AI-generated content may be incorrect.Тази функция се изпълнява безкрайно след изпълнението на setup(). Тук се опеределя сегашният етап спрямо стойността на променливата stage.

Фигура 17-Функцита loop

* + - 1. Arduino

**Като интегрирана среда за разработка (IDE) е използвана Arduino IDE.** Тя е предназначена за програмиране на микроконтролери и създаване на вградени системи, като основен фокус е върху лесната работа с Arduino платки и съвместими хардуерни платформи. Разработена от Arduino Project, тази среда предоставя удобен интерфейс и основни инструменти, които опростяват процеса на писане, компилиране и качване на програмен код директно върху микроконтролери.Arduino IDE използва език, базиран на C/C++, и позволява на разработчиците да създават приложения чрез т.нар. **скици (sketches)**. Тя поддържа широк набор от Arduino платки като Uno, Mega, Nano, както и други базирани на AVR, ARM или ESP32 микроконтролери.Средата предлага вградени функции за проверка на синтаксиса, компилация на кода и директно зареждане към устройството чрез USB. Arduino IDE включва и **сериѐн монитор**, който позволява лесно отстраняване на грешки и мониторинг на комуникацията между компютъра и микроконтролера в реално време.Arduino IDE предоставя възможност за добавяне на допълнителни **библиотеки**, които разширяват функционалността ѝ – например работа с дисплеи, сензори, модули за комуникация и др. Чрез **Board Manager** и **Library Manager** потребителите могат лесно да добавят нови платки и библиотеки.Също така, Arduino IDE е с отворен код и се поддържа от голяма общност, което улеснява сътрудничеството и споделянето на проекти. Независимо дали се създават малки проекти за автоматизация, роботи, IoT устройства или сензорни системи, Arduino IDE предоставя стабилна и достъпна среда за разработка, както за начинаещи, така и за напреднали потребители.



Фигура 18-Лого на Arduino

* + 1. АПИ

АПИ-то е изградено с помощта на Node.js, който от своя страна работ с express.js архитектура за обработка на POST заявките , mysql2 позволява да се използват MySql команди за съсдаване на заявки чиято информация бива превърната в JSON формат за улеснена прехвърляне към микроконтролера или панела.

* + - 1. Функции в кода

Основният код за АПИ-то се съхранява във файла index.js. В началото на файла се дефинират нужните библиотеки и модули:

* dotenv — за зареждане на конфигурация от .env файл;
* express — за създаване на уеб сървър и маршрути (routes);
* mysql2 — за връзка и работа с MySQL база данни;
* cors — за разрешаване на Cross-Origin Resource Sharing.

След зареждането на библиотеките се създава инстанция на приложението чрез express(). Активира се обработката на JSON заявки с express.json().  
Активира се CORS чрез “app.use(cors())”. За създаване на връзка с базата от данни се използва “mysql.createConnection” където се задават нужните данни за връзката. След свързването на база данни се създавата крайните точки:

A computer screen with text

AI-generated content may be incorrect.

Фигура 19-Примерна крайна точка

* Крайна точка за извличнае на продук по идентификационен номер

Крайната точка „/article/:id“ е от тип get и позволява на мястото на „:id” да се сложи цяло число по което да се вземе продукта с даденият идентификационен номер чрез следната SQL команда:“ SELECT \* FROM article WHERE id=?”

* Крайна точка за извличнае на продук по идентификационен номер

Крайната точка „/article/“ е от тип get и изпълнява SQL командата „SELECT \* FROM article“ за да изведе всички продукти.

* Крайна точка за извличнае на всички категории

Крайната точка „/category“ е от тип get и изпълнява SQL командата „SELECT id, name FROM category“ за да изведе идентификационния номер и името на всяка категория.

* Крайна точка за извличнае на всички продукти от категория

Крайната точка „/category/articles/:id “ е от тип get и позволява на мястото на „:id” да се сложи цяло число по което да се вземе продукта с даденият идентификационен номер на категория чрез следната SQL команда:

“SELECT article.id, article.name, article.price, article.weight from article

JOIN article\_category ON article.id = article\_category.id\_article

JOIN category ON article\_category.id\_category = category.id

WHERE category.id = ? OR parent\_id = ?”

* Крайна точка за извличнае на съставките на продукт

Крайната точка „ /article/ingredients/:id “ е от тип get и позволява на мястото на „:id” да се сложи цяло число по което да се вземат съставките на продукта по даденият му идентификационен номер, чрез следната SQL команда:

“ SELECT i.name

FROM ingredients AS i

JOIN article\_ingredient AS ai ON i.id = ai.id\_ingredient

WHERE ai.id\_article = ?“

* Крайна точка за получаване на алергените на продукт

Крайната точка „ /article/allergies/:id “ е от тип get и позволява на мястото на „:id” да се сложи цяло число по което да се вземат алергените на продукта по даденият му идентификационен номер чрез следната SQL команда:

“ SELECT al.name

     FROM article AS ar

     LEFT JOIN article\_ingredient AS ai ON ai.id\_article = ar.id

     LEFT JOIN ingredients AS i ON i.id = ai.id\_ingredient

     LEFT JOIN ingredients\_allergies AS air ON air.id\_ingredient = i.id

     LEFT JOIN allergies AS al ON al.id = air.id\_allergie

     WHERE ar.id = ? AND al.name IS NOT NULL“

* Крайна точка за вмъкване на поръчка

Крайната точка „/Orders“ е от тип POST и приема заявки в формат application/json, съдържащи полетата id\_table и id\_article. Получените стойности се записват в таблицата Orders чрез следната SQL заявка: „INSERT INTO orders (id\_table, id\_article) VALUES (?, ?)“.

* Крайна точка за извличнае на пръчките

Крайната точка „/Show/Orders“ е от тип get и извлича идентификационния номер на поръчката, идентификационния номер на масата, името,цената и теглото на продукта чрез следната SQL заявка:

„SELECT o.id AS OrderID, t.id AS TableId, a.name AS Name, a.price, a.weight

FROM orders AS o

JOIN article AS a ON a.id = o.id\_article

LEFT JOIN ttable AS t ON t.id = o.id\_table“.

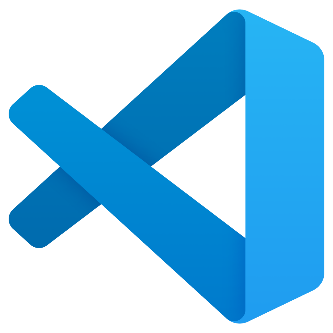
* Крайна точка за изтриване на поръчка

Крайната точка „/tables/order/:id“ е от тип DELETE и позволявана мястото на „:id“ да се постави идентификационния на поръчката която трябва да бъде премахната, това се осъществява с SQL заяката:

„DELETE FROM orders WHERE id = ?“

* + - 1. Visual Studio Code

В проекта е използвана средата за разработка **Visual Studio Code (VS Code)** – безплатен и лек текстов редактор, разработен от Microsoft, предназначен за програмиране на множество езици и технологии. VS Code предлага удобен и адаптивен интерфейс, поддръжка на разширения (extensions), интелигентно довършване на кода (IntelliSense), вградена терминална конзола и интеграция с Git.

Средата позволява лесно редактиране на HTML, CSS, JavaScript, Node.js и други езици, което я прави подходящ избор за уеб и сървърна разработка. В проекта VS Code е използван за писане и управление на сървърния код с **Express.js**, като благодарение на множеството налични плъгини, разработката е значително улеснена и ускорена.

Фигура 20-Лого на Visual Studio Code

* + - 1. Node JS

**Като среда за изпълнение и разработка на сървърни приложения е използвана Node.js.** Тя е предназначена за изграждане на високоефективни, мащабируеми уеб приложения и услуги, които работят от страна на сървъра. Node.js е базирана на JavaScript езика и използва V8 енджина на Google Chrome за бързо изпълнение на кода. Node.js е с отворен код и позволява на разработчиците да използват един и същ език – JavaScript – както за клиентската, така и за сървърната част на уеб приложенията. Това улеснява разработката и поддръжката на съвременни приложения. Средата е създадена с фокус върху производителност и ниска латентност, което я прави особено подходяща за реално време приложения като чатове, игри, уеб услуги и REST API-та. Node.js разполага с вграден **мениджър на пакети – npm (Node Package Manager)**, който предоставя достъп до огромен брой библиотеки и модули. Това позволява лесно добавяне на функционалности към проектите и ускорява процеса на разработка. Някои от най-често използваните библиотеки включват Express.js за създаване на уеб сървъри, Socket.io за реално време комуникация и Mongoose за работа с бази от данни MongoDB. Node.js се интегрира отлично с различни бази данни – както релационни (MySQL, PostgreSQL), така и нерелационни (MongoDB, Redis). Тя предлага **асинхронен и събитийно-ориентиран модел**, който позволява ефективна работа с множество заявки без блокиране на изпълнението. Съвместим с различни редактори и среди като **Visual Studio Code**, Node.js е често използван в екипни проекти с поддръжка на **контрол на версиите чрез Git**. Средата се използва широко в DevOps процеси, CI/CD pipelines и в съвременни облачни архитектури. Независимо дали става въпрос за разработка на микросървиси, уеб сървъри, API-та или приложения в реално време, **Node.js предоставя стабилна и модерна платформа**, която улеснява бързото и ефективно изграждане на надежден бекенд за съвременни уеб A group of logos on a black background

AI-generated content may be incorrect.системи.

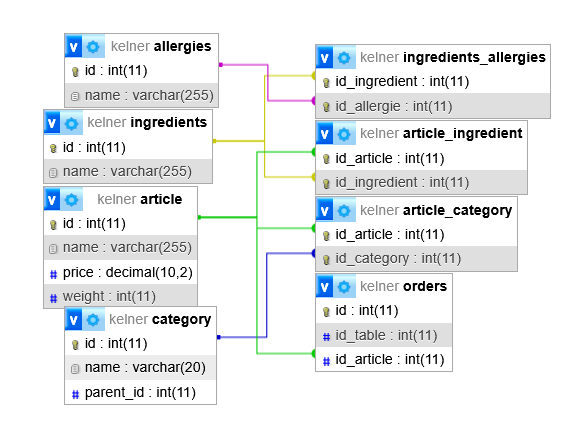
Фигура 21-Лого на Node.js

* + - 1. Express.js

За създаването на сървърната логика в приложението е използван **Express.js** – лек и гъвкав уеб фреймуърк за **Node.js**, предназначен за създаване на уеб приложения и REST API. Express опростява обработката на HTTP заявки, създаването на маршрути (routes) и управлението на middleware, като предоставя ясна и модулна структура за разработка.

В проекта Express.js служи като посредник между клиентската част и базата от данни, обработва заявки от потребителя, валидира данни и връща отговори в JSON формат. Благодарение на своята лекота и бързодействие, Express позволява бърза разработка на надежден сървър, който лесно се разширява с допълнителни функционалности като удостоверяване, логване и работа с файлове.

* + 1. Сървър

За хостване на сървър е използван XAMPP с MySql и phpMyAdmin за съхраняването на базата от данни.

Фигура 22- Схема на базата от данни

В таблица article се съдържа името цената ит теглото на продукта, като след това е свързана с таблици (от вида много към много с пирвичния клич на двете таблици които играят ролята на съставен първичн ключ): article\_ingrdients за съставките, article\_category за категорията на продукта и ingrdients\_allergies за алергийте на съставките на даден продукт.

Като в таблиците allergies,ingredients,article,category и orders има уникален индентификатор които игра ролята на първичен ключ.

В таблица orders биват запазени ID на поръчания пордукт, ID на масата, които биват получени от устроиството на масата.

Таблицата category има полета за идентификационен номер, име и parent\_id който се използва за разделение на главна категория и подкатегория.

* + - 1. XAMPP

**Като локална сървърна среда е използван XAMPP.** Тя е предназначена за разработка и тестване на уеб приложения в среда, симулираща реален уеб сървър. Създадена и поддържана от Apache Friends, тази платформа обединява в себе си няколко основни компонента, необходими за уеб разработка – Apache (уеб сървър), MySQL или MariaDB (сървър за бази данни), PHP и Perl (езици за програмиране). XAMPP е кросплатформена и може да бъде използвана както от Windows, така и от Linux и macOS. Благодарение на своята лесна инсталация и конфигурация, XAMPP е предпочитан инструмент от начинаещи и напреднали разработчици при изграждането и тестването на динамични уебсайтове и приложения, преди те да бъдат публикувани онлайн. Средата включва **контролен панел**, чрез който потребителите лесно могат да стартират и спират сървърите Apache и MySQL, както и да конфигурират различни настройки по услугите. XAMPP поддържа PHP скриптове и осигурява директна връзка с **phpMyAdmin** – графичен инструмент за управление на бази данни, който улеснява създаването, редактирането и администрирането на MySQL/MariaDB бази без необходимост от писане на SQL заявки. XAMPP е също така гъвкав при работа с различни уеб платформи и CMS системи като WordPress, Joomla, Drupal и други, които могат да бъдат инсталирани локално за разработка и тестване. Средата е съвместима с много разширения и конфигурационни модули, позволявайки разширяване на функционалността ѝ според нуждите на проекта. Независимо дали се изграждат малки уебсайтове, големи динамични системи или учебни проекти, XAMPP предоставя бърз и ефективен начин за локално разработване на приложения с реална сървърна инфраструктура, без необходимост от интернет връзка или хостинг услуги.

Фигура 23- Лого на XAMPP

* + - 1. phpMyAdmin

Като инструмент за управление на бази от данни е използван phpMyAdmin. Той е предназначен за администриране на релационни бази от данни MySQL и MariaDB чрез уеб базиран графичен интерфейс. Създаден с помощта на езика PHP, phpMyAdmin е един от най-популярните и използвани инструменти за работа с бази от данни в уеб среда. Благодарение на отворения си код и активната общност от разработчици, той се поддържа, обновява и разширява непрекъснато. phpMyAdmin предоставя удобен и интуитивен интерфейс, който позволява на потребителите да създават и управляват бази от данни, таблици, колони, индекси и записи. С него лесно могат да се изпълняват SQL заявки, да се филтрират и сортират данни, да се извършва търсене и да се преглеждат връзки между таблици. Инструментът предлага и широк набор от функции за експортиране и импортиране на данни в различни формати, включително SQL, CSV, Excel, XML и JSON. Благодарение на интегрираните възможности за управление на потребители и тяхната достъпност, phpMyAdmin позволява настройване на точни права и нива на сигурност за всяка база от данни. Поддържа също така SSL връзки, защита с парола и конфигурация на хостинг параметри за допълнителна сигурност. phpMyAdmin често се използва в комбинация със среди за локална разработка като XAMPP, WAMP и LAMP, което го прави незаменим инструмент за уеб разработчици и студенти, както и за професионалисти, занимаващи се с изграждане и поддръжка на динамични уеб сайтове и приложения. Той е особено полезен при работа по проекти, които изискват взаимодействие с база от данни, независимо дали става дума за регистрация на потребители, поръчки, съдържание или друга динамична информация. С phpMyAdmin се работи бързо и лесно, а възможността за визуализиране на структурата на базата и данните в нея спестява време и усилия на разработчиците. Независимо дали става въпрос за малки проекти или големи системи, phpMyAdmin предоставя надеждно и мощно решение за ефективно управление на бази от данни, като улеснява както тестването, така и поддръжката на приложения.

* + - 1. MySql

Фигура 24- Лого на phpMyAdmin

MySQL е многопоточна, многопотребителска, SQL система за управление на бази данни (СУБД) с повече от шест милиона инсталации.

MySQL AB разпространява MySQL като свободен софтуер под GNU General Public License (GPL), но също така под традиционните за комерсиален софтуер лицензи за случаи, когато използването е несъвместимо с GPL. Подобно разпространение е известно като двойно лицензиране.

MySQL се разработва, разпространява и поддържа от Шведската компания MySQL AB, която държи авторските права за голяма част от програмния код. Подобен е моделът на JBoss, а също и начинът, по който Free Software Foundation работи с авторските права по нейните проекти в отличие от Apache проекта, където софтуерът е обществено разработван, а авторските права за програмния код принадлежат на индивидуалните автори.

Компанията разработва и поддържа системата, продава поддръжка и сервизни договори, както и комерсиални лицензи за MySQL, и наема хора от целия свят, които работят съвместно с помощта на интернет. MySQL AB е основана от Давид Аксмарк, Алан Ларсон, и Михаел „Монти“ Видениус през 2001.

MySQL продава също и друга DBMS, MaxDB, която е с напълно различен код.

Достъпни са програмни интерфейси, позволяващи програми, написани на различни програмни езици да имат достъп до MySQL бази данни. Такива са: C, C++, C#, Delphi (чрез dbExpress), Eiffel, Smalltalk, Java (с директна поддръжка), Lisp, Perl, PHP, Python, Ruby, REALbasic (Mac), FreeBasic, и Tcl, като всеки от тях има специфичен програмен интерфейс. Интерфейс тип ODBC наречен MyODBC позволява на други програмни езици, които поддържат ODBC интерфейс да комуникират с MySQL база данни, например: ASP или Coldfusion. MySQL е написан основно на ANSI C.



Фигура 25- Лого на MySQL

* + 1. Визуалното приложение

A screenshot of a computer

AI-generated content may be incorrect.За визуализациая и управление на поръчките от страна на готвача е зползвано C# Winform приложение в което са визуализирани поръчките в ред на добовя, като най-отгоре е най новата.

Фигура 26- Визуалното приложение

* + - 1. Функции в кода
* Глобални променливи

int row = 0; - Запазва последната избраната поръчка

private BindingList<Order> orders; -Свързан списък за служещ за източник на данните на таблицата.

A screen shot of a computer code

AI-generated content may be incorrect.private Timer refreshTimer; - Създава таймер.

Фигура 27- Глобални променливи

* Конструкторът From1

A computer screen shot of a program code

AI-generated content may be incorrect.Конструкторът се извиква при стартиране на приложението и вътре се инициализират таблицата за приложението с функцията InitializeDataGridView(), HTTP клиента, източник на данни за таблицата, таймер и се извиквва функцията LoadOrdersAsync() за извличане на данните от HTTP клиента.

Фигура 28- конструктор

* Събитието RefreshTimer\_Tick()

Това събите се случва всяка секунда и е отговорно за обновяването на поръчките като първо извиква и изчаква функцията LoadOrdersAsync() и след отва подрежда таблицата по идентификатор на поръчката и запазва избрания ред.

* Функцията InitializeDataGridView()

Тази фунция служи за създаването таблицата, която има полета за идентификатор на поръчката, идентификатор на маса, име, цена и тегло на артикул.

* Функцията LoadOrdersAsync()

Тази Функция е отговорна за получаването на данните от АПИ-то. Тя спира временно обновяването на таблицата, след което прави връзка с крайната точка на АПИ-то, „/Show/Orders“ и ако е успешно направена връзката функцията получава JSON отговор който бива парсиран и предаден към нов списък „ordersList“. Свързаният списък „orders“ получава новите стойности от „ordersList“

A computer screen shot of a program code

AI-generated content may be incorrect.

Фигура 28- Функция LoadOrdersAsync

Фигура 29- Функция LoadOrdersAsync

* Събитие Delete\_Click

A screen shot of a computer code

AI-generated content may be incorrect.Това събитие се изпълнява когато се натисне бутона „DELETE“ и то изпраща идентификационния номер на поръчката която избрана за изтривавен към крайната точка „/tables/order/:id“ и премахва поръчката от свързния списък „orders“.

Фигура 30- Събитието Delete\_Click

* Събитие DeleteALL\_Click

A computer screen shot of a black screen

AI-generated content may be incorrect.Това събитие се изпълнява когато се натисне бутона „DELETE“ и то използва цикъл, за да изпрати идентификационния номер на всяка една поръчка към крайната точка „/tables/order/:id“

Фигура 31- Събитието DeleteALL\_Click

* Класът Order

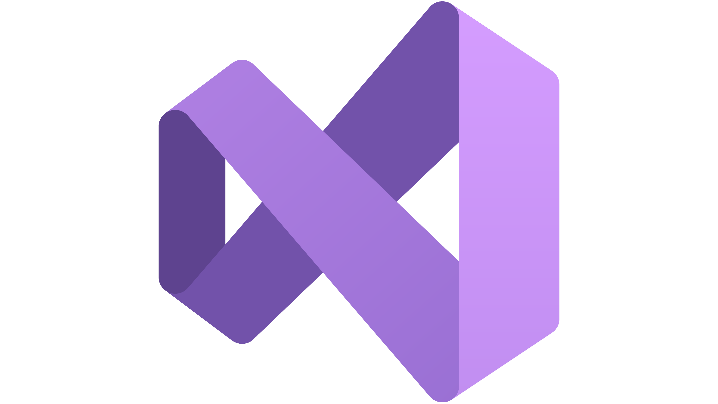
Този клас слъжи служи за създаванен на нов тип данни за получените данни от АПИ-то.

* Функциите Up\_Click и Down\_Click

Тези функции променят стойността на променливата за избрания ред(row) при натискане на съответния бутон.

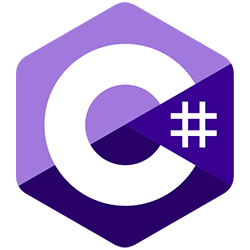
* + - 1. Visual Studio Community 2022

Като интегрирана среда за разработка (IDE) е използвана Visual Studio. Предназначена е за разработка на софтуерни приложения в широк спектър от технологии и платформи. Създадена от Microsoft, тази мощна платформа предоставя обширен комплект инструменти и функции, които улесняват и ускоряват процеса на създаване на софтуер. С Visual Studio разработчиците могат да работят с различни езици за програмиране като C#, C++, Visual Basic, F# и други. Той предлага интегрирана поддръжка за различни технологии и платформи, включително уеб разработка с ASP.NET и HTML/CSS/JavaScript, създаване на десктоп приложения с Windows Forms или WPF, мобилни приложения за Android и iOS, управление на бази от данни с SQL Server и много други. Visual Studio е познат също така със своите мощни инструменти за отстраняване на грешки (debugging), профилиране на приложенията, автоматизирано тестване и управление на източници. Той предлага интегрирана система за контрол на версиите, която улеснява работата на екипите от разработчици, като им позволява да си сътрудничат по ефективен начин. С Visual Studio разработчиците могат да използват разширения и добавки, които допълват функционалността на платформата. Тези разширения включват инструменти за улеснено сътрудничество, подобрено управление на проекти, разширена поддръжка за определени технологии и други. Без значение дали става въпрос за създаване на уеб приложения, десктоп софтуер, мобилни приложения или други видове софтуерни проекти, Visual Studio предоставя надежден и мощен набор от инструменти, които помагат на разработчиците да постигнат успех в техните усилия.



Фигура 32- Логото на Visual Studio Community

* + - 1. C# Forms

За разработката на приложението е използвана технологията Windows Forms (C# Forms), интегрирана в средата за разработка Visual Studio. Тя е предназначена за създаване на настолни приложения с графичен потребителски интерфейс (GUI) в среда на Windows. Разработена от Microsoft, тази технология предоставя лесен и интуитивен начин за изграждане на богати на функционалности потребителски интерфейси чрез влачене и пускане на елементи от формата (drag-and-drop). Windows Forms е част от .NET Framework и .NET Core/.NET 5+ платформите и позволява използването на езика C# за логиката на приложенията. С тази технология разработчиците могат да създават форми (windows), да добавят бутони, текстови полета, списъци, менюта и други елементи на интерфейса, като същевременно имат достъп до пълната мощ на .NET библиотеките и обектно-ориентираното програмиране. Windows Forms се интегрира напълно във Visual Studio, което позволява визуално проектиране на интерфейса, автоматично генериране на код за контролите и бързо свързване с обработчици на събития. Visual Studio предоставя и инструменти за отстраняване на грешки, профилиране, рефакториране и управление на проекта, което значително ускорява разработката.С помощта на Windows Forms разработчиците могат лесно да свързват своите приложения с бази от данни, използвайки ADO.NET или Entity Framework. Поддръжката на визуални контроли и лесната обработка на събития прави Windows Forms подходящ избор за създаване на административни панели, инструменти за управление, информационни системи и други десктоп приложения. Visual Studio позволява и интеграция с системи за контрол на версиите, както и използване на външни разширения, които добавят допълнителна функционалност към процеса на разработка. Независимо дали става дума за малки помощни програми или по-сложни бизнес приложения, Windows Forms с C# е стабилна и надеждна платформа за бързо създаване на Windows-базирани решения.

Фигура 33- Логото на C#

1. Инсталация и поддръжка

Изивсквания за инсталация:

* Компютър с инсталиран Node.js и XAMPP
* Локална безжична мрежа

Стъпки на инсталация

1. В XAMPP Control Panel се стартира Apache и MySql
2. Отваря се phpMyAdmin чрез Admin бутона на MySql в XAMPP Control Panel
3. Импортира се базата от данни и се правят съответните промени по менюто
4. Папка KelnerAPI се отваря в Visual Studio Code
5. В index.js се променят константата db се променя такаче да отговаря на базата от данни
6. За да се стартира АПИ-то се отваря терминала за папка KelnerAPI и се пуска следната команда: npm start. Ако всичко направено правилно трябва да се изпиши :

Server running at <http://0.0.0.0:5000>

Connected to MySQL

1. В Display/Display.ino се променят SSID и Password такаче да отговарят на безжичната мрежа,като зависимост от мрежата може да трябва да се премахне WiFi.mode(WIFI\_STA), след това се променя table\_id да отговаря номера на масата и накрая се променя baseAPI такаче да отговаря на IPv4 на машината на която е хостван сървърът.
2. Преди качването на кода на микроконтролера нужно да е подсигурено че устройството не е включено от батерията за да не се повреди системата.
3. За да се качи кода на микроконтролера трябва да се задържи бутнот отляво на USB type-C порта и да се натисне лявото зада се включи Download мода на платката
4. Стартираме kelnercho.exe приложението намиращо се в Winform\Kelnercho\bin\Release на сървърния компютър
5. Бъдеща реализация

Въпреки че системата може да бъде приложена в реална среда, тя може да бъде надградена с допълнителни функционалнсики като:

* Съхраняване на сметка със поръчаните продукти на маса.
* Свързване с ПОС терминал за разплащане
* Подобряване и улесняване на контролът на менюто
* Справка за количеството съставки за продукт и ако няма достатъчно количество съставки продукта временно да бъде премахант от менюто
* Добавяне на български и други езици
* Добавянен на начин за следене на статуса на поръчката

1. Заключение

Келнер системата представлява комбинация от хардуер и софтуер за оптимизиране, подсигуряване и забързване на процеса на поръчване в ресторанти, заведения за бързо хранене и кафенета. Основанието за разработката на системата идва от лисата на подобно интегрирано решение в индустрията, което да обединява лесен и интуитивен интерфейс с централно управление на поръчките. Целевата аудитория на системата са както и клиентите на заведението така и персоналът в кухнята.

1. Използвана литература

За изработката на системата са използвани:

* <https://dev.to/anticoder03/building-restful-apis-with-nodejs-and-express-step-by-step-tutorial-2oc6>
* <https://randomnerdtutorials.com/esp32-tft-touchscreen-display-2-8-ili9341-arduino/>
* Stack Overflow
* <https://www.newtonsoft.com/json/help/html/Introduction.htm>
* <https://learn.microsoft.com/en-us/dotnet/api/system.net.http?view=net-9.0>