# **Projektterv 2024-2025**

## **1. Összefoglaló**

A projekt lényege, hogy egy vásárló távolságtól függően fel tudja venni a kapcsolatot egy pizzériával és kéréseket tudjon adni a pizzéria felé. Ez tipikusan megkövetel egy online is elérhető nyilvántartást, ahol rugalmasan és egyszerűen lehet kezelni a kapcsolódó feladatokat. Éppen ezért ennek a projektnek a célja, hogy egy pizzéria számára egy webalkalmazást fejlesszünk, amely képes leváltani a hagyományos rendelési és kapcsolatfelvétel módszereket. Legyen szó rendelésről, asztalfoglalásról vagy panaszbejelentésről a cél, hogy a kapcsolódó funkciókat egy letisztult felületen keresztül tudja elérni a pizzéria vásárlói.

## **2. A projekt bemutatása**

Ez a projektterv a Pizza Torony projektet mutatja be, amely 2024-10-22-től 2025-04-30-ig tart. A projekten három fejlesztő fog dolgozni, az elvégzett feladatokat pedig 6 alkalommal fogják prezentálni a megrendelőnek, annak érdekében, hogy biztosítsuk a projekt folyamatos előrehaladását.

### **2.1. Rendszerspecifikáció**

A projekt fő célja egy intuitív és reszponzív weboldal létrehozása ételek rendelésére és vásárlása. Lennie kell egy áruháznak, ami megmutatja a rendelhető terméket és azokról releváns információkat (pl. név, ár, leírás, leárazások, stb.). A kiválasztott termékeket egy kosárba teszi és egy kitöltendő formán keresztül a felhasználó megtudja rendelni az kosárban lévő termékeket. A weboldalon lennie kell olyan részeknek, amik a pizzériával kapcsolatos releváns információkat fogják megjeleníteni (pl. hírek, adatok kapcsolatfelvételhez, nyitvatartási idők, stb.). Kitöltendő formákon keresztül a felhasználó fel tudja venni a kapcsolatot az étteremmel egy panasz esetén és tud előre asztalt foglalni.

### **2.2. Funkcionális követelmények**

* Felhasználók kezelése (CRUD)
* Ételek kezelése (CRUD)
* Kosarak kezelése (CRUD)
* Asztalfoglalások kezelése (CRUD)
* Rendelések kezelése (CRUD)
* Üzenetek kezelése (CRUD)

A CRUD rövidítés a következőt jelenti: Create, Read, Update, Delete.  
Azaz a funkcióhoz tartozó adat létrehozása, olvasása, frissítése, törlése.

### **2.3. Nem funkcionális követelmények**

* A kliens oldal böngészőfüggetlen legyen
* Reszponzív megjelenés (optimális megjelenés különfélébb eszközökön)
* Az adatokat MySQL adattáblákban tároljuk
* A legfrissebb technológiákat használja a rendszer

## **3. Projektcsapat**

| Név | Pozíció | E-mail cím |
| --- | --- | --- |
| Pontos Elek | Projektmenedzser | pontos.elek-2020@keri.mako.hu |
| Remek Elek | Projekt tag | remek.elek-2020@keri.mako.hu.hu |
| Teszt Elek | Projekt tag | teszt.elek-2020@keri.mako.hu |

## 

## **4. A munka feltételei**

### **4.1. Munkakörnyezet**

A projekt a következő technológiákat/szoftvereket fogja használni a munka során:

* Apacs szerver a webalkalmazás hosztolásához
* MySQL adatbázisszerver, phpMyAdmin
* Programozási nyelvek/keretrendszerek:
  + FRONTEND: Html5, Css3, JavaScript (ES6), Bootstrap, jQuery, Angularjs

(opcionális: Angular, React, Vue, …)

* + BACKEND: Php (opcionális: Laravel, Python, NodeJS, …)
* Git verziókövető (GitHub)
* IDE – Visual Studio Code

### **4.2. Rizikómenedzsment**

| Kockázat | Leírás | Valószínűség | Hatás |
| --- | --- | --- | --- |
| Betegség | Súlyosságtól függően hátráltatja vagy bizonyos esetekben teljes mértékben korlátozza a munkavégzőt, így az egész projektre kihatással van. Megoldás: a feladatok átcsoportosítása | nagy | erős |
| Kommunikációs fennakadás a csapattagokkal | A csapattagok között nem elégséges az információ áramlás, nem pontosan, esetleg késve vagy nem egyértelműen tájékoztatjuk egymást. Megoldás: még gyakoribb megbeszélések és ellenőrzések | kis | erős |
| szoftver-hardver probléma | Egy nem várt szoftver/hardver hiba lép fel, ami korlátozza a projekt folytatását. Megoldás: a hibát megkeresni és kijavítani. Ha a hibát nem lehet kijavítani akkor egy másik szoftveren/hardveren kell folytatni a feladatokat vagy ki kell cserélni a szoftvert/hardvert. | nagy | erős |
| tag kiesése | A projekt egyik tagja már nincs a csapatban. Megoldás: a feladatok átcsoportosítása és, ha lehetséges egy új tag felvétele. | kis | erős |

## 

## **5. Jelentések**

### **5.1. Munka menedzsment**

A munkát Pontos Elek koordinálja. Fő feladata, hogy folyamatosan egyeztessen a csapattagokkal az előrehaladásról és a fellépő problémákról, esetlegesen a megoldásban is segítséget nyújthat a projekt csúszásának elkerülése végett. További feladata a heti szinten tartandó csoportgyűlések időpontjának és helyszínének megszervezése.

### **5.2. Csoportgyűlések**

A projekttagok hetente megvitassák az azt megelőző hét problémáit, illetve, hogy megbeszéljék a következő hét feladatait. A megbeszélésről minden esetben memó készül.  
(időpont, hely, résztvevők, érintett témák, összefoglalás, …)

### **5.3. Minőségbiztosítás**

Az elkészült terveket a terveken nem dolgozó csapattársak átnézik, hogy megfelel-e a specifikációnak és az egyes diagramtípusok összhangban vannak-e egymással. A meglévő rendszerünk helyes működését a prototípusok bemutatása előtt a tesztelési dokumentumban leírtak végrehajtása alapján ellenőrizzük és összevetjük a specifikációval, hogy az elvárt eredményt kapjuk-e. További tesztelési lehetőségek: unit tesztek írása az egyes modulokhoz vagy a kód közös átnézése (code review) egy, a vizsgált modul programozásában nem résztvevő csapattaggal. Szoftverünk minőségét a végső leadás előtt javítani kell a rendszerünkre lefuttatott kódelemzés során kapott metrikaértékek és szabálysértések figyelembevételével. Az alábbi lehetőségek vannak a szoftver megfelelő minőségének biztosítására:

* Specifikáció és tervek átnézése (kötelező)
* Teszttervek végrehajtása (kötelező)
* Unit tesztek írása (választható)
* Kód átnézése (választható)

### **5.4. Státuszjelentés**

Minden mérföldkő leadásnál a projekten dolgozók jelentést tesznek a mérföldkőben végzett munkájukról. A csapatmegbeszéléseken a csapattagok áttekintik és felmérik az eredményeket és teendőket. Továbbá gazdálkodnak az erőforrásokkal és szükség esetén a megrendelővel egyeztetnek a projektterv módosításáról.

## **6. A munka tartalma**

### **6.1. Tervezett szoftverfolyamat modell és architektúra**

A szoftver fejlesztése során az agilis fejlesztési modellt alkalmazzuk, mivel a fejlesztés során nagy hangsúlyt fektetünk a folyamatos kommunikációra. A fejlesztés során a szoftver specifikációi rugalmasan változhatnak, és ezzel a módszertannal tudunk a leggyorsabban alkalmazkodni az új elvárásokhoz. A webalkalmazás az MVC (modell-view-controller) felépítést követi, a szerver és a kliens függetlenek, csupán Ajax API végpontok segítségével kommunikálnak.

### **6.2. Átadandók és határidők**

A főbb átadandók és határidők a projekt időtartama alatt a következők:

| Szállítandó | Neve | Határideje |
| --- | --- | --- |
| D1 | Projektterv, egyéni jelentések | 2024-11-14 |
| P1+D2 | UML, adatbázis- és képernyőtervek, egyéni jelentések | 2024-12-12 |
| P1+D3 | Prototípus I., tesztelési dokumentáció, egyéni jelentések | 2025-01-16 |
| P2+D4 | Prototípus II., frissített tesztelési dokumentáció, egyéni jelentések | 2025-02-13 |
| P3+D5 | Prototípus III., frissített tesztelési dokumentáció, egyéni jelentések | 2025-03-20 |
| B+D6 | Termék béta verzió, frissített tesztelési dokumentáció, egyéni jelentések | 2025-04-16 |
| V+D7 | Végtermék, és a teljes dokumentáció | 2025-04-30 |

**D** - dokumentáció,   
**P** – prototípus  
**B** – termék béta verzió  
**V** - végtermék

## **7. Részletes feladatlista**

A feladatok megszervezésére, a **Trello** projektmenedzsment szoftvert használjuk.  
A teljes projektet a projektmenedzser koordinálja.  
A kártyák reprezentálják az egyes feladatokat, melyeket listákba csoportosítunk:

* teendők,
* kiosztva, folyamatban,
* befejezve,
* dokumentálva,
* tesztelve,
* véglegesítve.

Minden egyes feladatnak (kártyának) rendelkeznie kell:

* felelős személy
* munkát elvégző személy(ek)
* tervezett határidő
* megvalósítás ideje

## **8. Csatolt dokumentumok**

### **8.1. Fejlesztői dokumentáció**

* A fejlesztőkörnyezet (hardver és szoftverek) ismertetése, és a választás indoklása.
* Adatmodell (UML diagram), a kialakított adatszerkezet részletes bemutatása.
* Felhasználói felület (UI) tervek (Figma).
* A fejlesztés fázisainak leírása (Trello).
* Teszt dokumentáció.
* A szoftver lehetséges továbbfejlesztésének tervezete.

### **8.2. Felhasználói dokumentáció**

* A program céljának, és lényegesebb funkcióinak összefoglalása.
* A program részletes bemutatása (funkcionális).
* Képernyőképek készítése, magyarázata.

## **9. Irodalomjegyzék**

* W3School  
  <https://www.w3schools.com/>
* Bootstrap 5  
  <https://getbootstrap.com/>
* Unsplash  
  <https://unsplash.com/>
* ChatGPT  
  <https://openai.com/chatgpt/overview/>