Projekt munka feladatok

Felhasználói nézet

1. **Bejelentkező felület / Regisztrációs felület**
   1. *DataBase:* Egy olyan személy tábla készítése, amely tárol: id-t (pmk), vezetéknevet, keresztnevet, email címet, lakcímet (bontva: irsz, telep, cím), telefonszámot. Továbbá kell egy bejelentkezési tábla, amely ID azonosan tárol jelszót.
   2. *Backend:* Megvalósítja a regisztrációs szinkronizációt, azaz a felhasználó regisztrál, az bekerül a személy táblába, illetve ID azonosan kerül be a jelszó a másik táblába. Bejelentkezéskor pedig ellenőrzi az email cím és jelszó párost. Ha valami hibát vagy hiányzó adatot észlel hibeüzenettel tér vissza. Megfelelő hivatkozásokkal látja el az oldalt: reg -> bejelentkezés, bejelentkezés -> kezdőlap
   3. *Frontend:* Megvalósítja a regisztrációs, illetve bejelentkezési oldalt. Regisztrációs és bejelentkezési oldalon is a megfelelő adatokat bekéri. Ügyel arra, hogy a jelszó ne karakterekben látszódjon, hanem hidden-ként(pontok). Továbbá felkészíti az oldalt arra, hogy hiányzó adatok esetén milyen hibaüzenet jön vissza – backenddel együtt dolgozva. Megfelelően kialakítja a gombokat.
2. **Kezdőlap fül**
   1. *Frontend*: A felhasználó a kezdőlapon a következő fülekre tud rámenni, illetve legelsőként valami kis üdvözlő felirat fogadja. Oldalt tudjon választani a következőek közül:
      1. Új probléma felvétele
      2. Bejelentéseim
      3. Üzenetek
      4. Kapcsolat
      5. Hírek
   2. *Backend*: A megfelelő fülre való hivatkozásokat készíti el.
3. **Új probléma felvétele fül**
   1. *DataBase*: Olyan tábla kialakítása, amely tárolja: probléma az (egyedi pmk), a probléma észlelésének helyét (legördülő listából), időpontját (akt. dátum és időpont), képeket képes fogadni, egyéb leírás a pontosítás miatt, illetve egy olyan mezőt amit a felhasználó nem lát, hogy kié a feladat. Továbbá az adatbázisban legyen olyan mező/mezők, amelyek a hiba 3 fázisát jelölik: Felvéve, Folyamatban, Kész – ezeket az adminisztrátor tudja módosítani. Legyen egy kapcsoló tábla, amely a bejelentések azonosítóját, illetve a felhasználó azonosítóját párosítja.
   2. *Backend*: Az űrlap megfelelő beviteli mezőit szinkronizálja az adatbázisba, illetve azokat az adatokat, amelyeket automatikusan tölt ki az űrlap - elküldi az adatbázisból. A beküldés után a megfelelő üzenettel tér vissza.
   3. *Frontend*: Megtervezi az űrlapot, amelyben szerepeljen az a rész, hogy ki a bejelentő – azaz a felhasználó de ez majd automatikusan töltődik ki, fontos jelenjen meg a saját azonosítója is. Továbbá kell egy bejelentés helye, amelyhez egy legördülő listából lehet választani. A probléma észlelésének időpontja (aktuális dátum, időpont), képek, illetve egyéb kifejtős rész. Az elküldés után jelzi annak sikerességét az oldal – ezt backenddel közösen valósítja meg – megjeleníti az adott probléma azonosítóját.
4. **Bejelentéseim fül:**
   1. *DataBase*: Az adott id felhasználó bejelentéseihez megírja a lekérdezést.
   2. *Backend*: A kapcsoló tábla alapján lekéri az adott felhasználóhoz tartozó bejelentéseket és továbbítja.
   3. *Frontend*: Megfelelő design mellett az adott felhasználóhoz tartozó bejelentéseket adja vissza. Ami fontos az alapadatok mellett szerepeljen egy jelzőbox/csúszka ami mutatja, hogy épp milyen fázisban van a bejelentése.
5. **Üzenetek fül**
   1. Új üzenet kezdése
      1. *DataBase*: Létrehoz egy üzenetek táblát, melynek mezői a következők: Személy (Küldő) ID, Címzett ID (munkatársak táblából – legördülő lista), Üzenet ID (aut. gen.), Tárgy (legördülő lista), Üzenet – olvasási visszaigazolás legyen!
      2. *Backend*: Biztosítja a szükséges szinkronizációt, illetve a személy ID-t automatikusan hívja le – olvasási visszaigazolás.
      3. *Frontend*: Megtervezi az üzenetküldés mezőit, személy azonosítója automatikusan íródik be.
   2. Üzeneteim
      1. *DateBase*: Lekérdezést ír az üzenetek táblából (későbbiekben az adminisztrátor üzenetei közül is kiolvassa) – személy ID alapján
      2. *Backend*: Megvalósítja a lekérdezést, biztosítja a levél olvashatóságát
      3. *Frontend*: Vizualizálja a lekérdezést, levélre kattintva a levél nagyban olvasható
6. **Kapcsolat fül**
   1. *Frontend*: A megfelelő hivatkozásokat, ügyfél fogadási időt, irodát, telefonszámot teszi ide.
7. **Hírek fül:**
   1. DataBase: Hírek táblából olvassa ki az adatokat (Hír ID, Cím, Létrehozás dátuma, tartalma)
   2. Backend: Biztosítja a lekérdezés általi kiolvasást.
   3. *Frontend*: Hírek hivatkozásait teszi ide egy egy fotóval, valamilyen közösségi média platformról

Osztályvezető/főnök felület

1. **Bejelentkezés – regisztráció nincs, mivel a rendszergazda veszi majd fel**
   1. DataBase: Létrehozunk egy alkalmazotti táblát, amelybe szerepel az alkalmazott: ID, vezetéknév, keresztnév, email cím, telefonszám. Külön lesz itt is egy password tábla amely ID-val legyen összekapcsolva.
   2. Backend: A megfelelő adatok ellenőrzése után tovább engedi a kezdőlapra az alkalmazottat.
   3. Frontend: Bejelentkezési boxok és felület megtervezése: Email, jelszó
2. **Kezdőlap**
   1. Backend: Hivatkozások kezelése
   2. Frontend: A következő fülek megtervezése:
      1. Bejelentések
      2. Üzenetek
      3. Hírek
3. **Bejelentések**
   1. DateBase: A korábban már ehhez megtervezett adatbázisból tesz lekérdezéseket, illetve van jogosultsága a szerkesztéshez, illetve képes állatítani egy feladat állapotát, meghatározni kihez tartozik
   2. Backened: A szükséges lekérdezéseket alkalmazza. Biztosítja az íráshoz való hozzáférést, majd a mentéskori szinkronizációt.
   3. Frontend: A bejelentéseket látja, illetve képes ezekbe belenyúlni, majd menteni.
4. **Üzenetek**
   1. Új üzenet írása
      1. Backend: Biztosítja az új levél szinkronizációját, illetve a címzett azonosítójának beírása után a levél elküldését
      2. Frontend: Megtervezi a levél küldésének felületét.
   2. Üzenetek
      1. DataBase: Megírja a lekérdezést, vagyis úgy írja meg, hogy az osztályvezető képes legyen a levelekre szűrni. Mindegyik levelet láthatja
      2. Backend: biztosítja a lekérdezést és szűrést. Levélre kattintás esetén megnyílik a levél tartalma.
      3. Frontend: megtervezi a levelek nézegetős felületét. Megtervezi a levél olvasó nézetet is, amikor a levél nagyban látszódik.
5. **Hírek**
   1. Hírek
      1. DataBase: Lekérdezést ír.
      2. Backend: Lekérdezést végrehajtja, hivatkozási kattintás után a hírt nagyban adja ki.
      3. Frontend: A híreket olvasható formában mutatja, illetve a hivatkozási kattintást
   2. Új hír létrehozása
      1. Backend: Megfelelő szinkronizáció, illetve az adatbázisba való létrehozás
      2. Frontend: A megfelelő mezőkkel/boxokkal létrehozza a hír írási felületet.

Alkalmazotti felület

1. **Bejelentkezés – regisztráció nincs, mivel a rendszergazda veszi majd fel**
   1. DataBase: Létrehozunk egy alkalmazotti táblát, amelybe szerepel az alkalmazott: ID, vezetéknév, keresztnév, email cím, telefonszám. Külön lesz itt is egy password tábla amely ID-val legyen összekapcsolva.
   2. Backend: A megfelelő adatok ellenőrzése után tovább engedi a kezdőlapra az alkalmazottat.
   3. Frontend: Bejelentkezési boxok és felület megtervezése: Email, jelszó
2. **Kezdőlap**
   1. Backend: Hivatkozások kezelése
   2. Frontend: A következő fülek megtervezése:
      1. Bejelentések
      2. Üzenetek
      3. Hírek
3. **Bejelentések**
   1. DateBase: A korábban már ehhez megtervezett adatbázisból lekérdezés alapján azt jeleníti meg, ami a kolléga hatásköre, illetve van jogosultsága egy feladat állapotát módosítani
   2. Backened: A szükséges lekérdezéseket alkalmazza. Biztosítja az íráshoz való hozzáférést, majd a mentéskori szinkronizációt.
   3. Frontend: A bejelentéseket látja, illetve képes ezekbe belenyúlni, majd menteni.
4. **Üzenetek**
   1. Új üzenet írása
      1. Backend: Biztosítja az új levél szinkronizációját, illetve a címzett azonosítójának beírása után a levél elküldését
      2. Frontend: Megtervezi a levél küldésének felületét.
   2. Üzenetek
      1. DataBase: Megírja a lekérdezést, hogy csak a saját leveleit látja.
      2. Backend: biztosítja a lekérdezést és szűrést. Levélre kattintás esetén megnyílik a levél tartalma.
      3. Frontend: megtervezi a levelek nézegetős felületét. Megtervezi a levél olvasó nézetet is, amikor a levél nagyban látszódik.
5. **Hírek**
   1. Hírek
      1. DataBase: Lekérdezést ír.
      2. Backend: Lekérdezést végrehajtja, hivatkozási kattintás után a hírt nagyban adja ki.
      3. Frontend: A híreket olvasható formában mutatja, illetve a hivatkozási kattintást

Rendszergazdai felület

1. **Bejelentkezés – regisztráció nincs, mivel a rendszergazda veszi majd fel**
   1. DataBase: Létrehozunk egy rendszergazda táblát, amelybe szerepel az alkalmazott: ID, vezetéknév, keresztnév, email cím, telefonszám. Külön lesz itt is egy password tábla amely ID-val legyen összekapcsolva.
   2. Backend: A megfelelő adatok ellenőrzése után tovább engedi a kezdőlapra az alkalmazottat.
   3. Frontend: Bejelentkezési boxok és felület megtervezése: Email, jelszó
2. **Kezdőlap**
   1. Backend: Hivatkozások kezelése
   2. Frontend: A következő fülek megtervezése:
      1. Bejelentések
      2. Üzenetek
      3. Hírek
      4. Új ember felvétele
3. **Bejelentések**
   1. DateBase: A korábban már ehhez megtervezett adatbázisból tesz lekérdezéseket, illetve van jogosultsága a szerkesztéshez, illetve képes állatítani egy feladat állapotát, meghatározni kihez tartozik
   2. Backened: A szükséges lekérdezéseket alkalmazza. Biztosítja az íráshoz való hozzáférést, majd a mentéskori szinkronizációt.
   3. Frontend: A bejelentéseket látja, illetve képes ezekbe belenyúlni, majd menteni.
4. **Üzenetek**
   1. Új üzenet írása
      1. Backend: Biztosítja az új levél szinkronizációját, illetve a címzett azonosítójának beírása után a levél elküldését
      2. Frontend: Megtervezi a levél küldésének felületét.
   2. Üzenetek
      1. DataBase: Megírja a lekérdezést, vagyis úgy írja meg, hogy a rendszergazda képes legyen a levelekre szűrni. Mindegyik levelet láthatja
      2. Backend: biztosítja a lekérdezést és szűrést. Levélre kattintás esetén megnyílik a levél tartalma.
      3. Frontend: megtervezi a levelek nézegetős felületét. Megtervezi a levél olvasó nézetet is, amikor a levél nagyban látszódik.
5. **Hírek**
   1. Hírek
      1. DataBase: Lekérdezést ír.
      2. Backend: Lekérdezést végrehajtja, hivatkozási kattintás után a hírt nagyban adja ki.
      3. Frontend: A híreket olvasható formában mutatja, illetve a hivatkozási kattintást
   2. Új hír létrehozása
      1. Backend: Megfelelő szinkronizáció, illetve az adatbázisba való létrehozás
      2. Frontend: A megfelelő mezőkkel/boxokkal létrehozza a hír írási felületet.
6. **Új ember felvétele**
   1. DataBase: A megfelelő adatbázishoz adja hozzá a dolgozót, vagy felhasználót
   2. Backend: Biztosítja a szinkronizációt
   3. Frontend: Megtervezi a felvevő felületet