

Technikai dokumentáció az IT School oldalhoz

Felhasznált nyelvek, keretrendszerek



HTML5



CSS



Bootstrap
Bootstrap



Java Script



php



AngularJS



MySQL



jQuery

Minden oldalon található Header és Footer amik CSS-s és Bootstrap osztályokkal lettek megformázva

```
.courses-title {
  background-color: rgba(0,0,0,0.5);
}

.navbar, .footer{
  background-image: linear-gradient(to right, rgb(48, 50, 163) , rgb(103, 104, 173));
}

.nav-item{
  list-style-type: none;
  display: inline-block;
  margin-right: 20px;
}
```

```
<!-- Footer -->
<div id="availability" class="container-fluid pt-3 footer">

  <!-- School -->
  <div class="row footer-bg">

    <!-- Logo -->
    <div class="col-3 col-xl-1">
      <div class="logo bg-img bg-img-contain float-end"></div>
    </div>

    <!-- Name -->
    <ul class="col-9 col-xl-5 text-start list-unstyled mb-0 mt-1
      foot-item text-capitalize">
      <li>
        <h5>
          {{'school' | translate:lang.data}}
        </h5>
      </li>
    </ul>
  </div>
</div>
```

```
<!-- Navbar -->
<nav class="navbar navbar-right navbar-expand-lg">

  <!-- Container -->
  <div class="container-fluid">

    <!-- Logo -->
    <a class="nav-link"
      ng-class="{ 'disabled': state.id==='user' }"
      ui-sref="home">
      
    </a>

    <!-- Hamburger icon -->
    <button class="navbar-toggler bg-light"
      type="button"
      data-bs-toggle="collapse"
      data-bs-target="#navbarNav"
      aria-controls="navbarNav"
      aria-expanded="false"
      aria-label="Toggle navigation">
      <span class="navbar-toggler-icon bg-light"></span>
    </button>
```

Az oldal linkek
ng-class
használatával kapnak
formázást kijelöléskor,
illetve amikor aktívak

```
<!-- Navigate -->
<ul class="navbar-nav">

  <!--Home-->
  <li class="nav-item">
    <a class="nav-link text-lowercase mt-2"
      ng-class="{ 'active cursor-default text-decoration-underline
      | fw-bold fs-6 text-uppercase': state.id === 'home' }"
      ui-sref="home">
      <span>
        <p class="hover-underline-animation navfont">
          {{ 'home_page' | translate:lang.data }}
        </p>
      </span>
    </a>
  </li>
```

Minden, az oldalon található szöveg a megfelelő nyelvi json
fájból kerül kiolvasásra kulcsok segítségével.

```
"subjects1": "Design und Entwicklung von autonomen Fahrzeugen",
"subjects2": "Robotik und Softwareentwicklung für Fahrzeuge",
"subjects3": "Fahrzeugbetrieb und Flottenmanagement",
"subjects4": "Künstliche Intelligenz und maschinelles Lernen in Fahrzeuganwendungen",
"subjects5": "Fahrzeugsicherheit und rechtliche Fragen im Bereich der selbstfahrenden Fahrzeuge",
"subjects6": "Grundlagen der Smart-Home-Technologien",
"subjects7": "Intelligente Netze und Kommunikation",
"subjects8": "Entwurf und Integration eines intelligenten Heimsystems",
"subjects9": "Automatisierte Haussicherheit",
"subjects10": "Smart Home Management und Energieeffizienz",
```

```
▼ lang
  {} available.json
  {} de.json
  {} en.json
  {} hu.json
```

Ezek a kulcsok a translate filter használatával kerülnek meghívásra.

```
<!-- Title -->
<h1 class="text-center text-capitalize text-small-caps">
  {{course.course_name | translate:lang.data}}
</h1>
<hr>
</div>

<div class="col-12 col-md-10 col-lg-8 mx-auto text-center">
  <hr>
  <p>{{course.description | translate:lang.data}}
  <hr>
</p>
</div>
```

```
// Translate
.filter('translate', [
  'util',
  (util) => {
    return (key, data=null, isAllowed=true) => {
      if (!util.isString(key)) return key;
      key = key.trim();
      if (util.isString(isAllowed))
        isAllowed = !(isAllowed.toLowerCase().trim() === 'false');
      if (!util.isBoolean(isAllowed))
        isAllowed = true;
      if (!isAllowed ||
          !util.isObjectHasKey(data, key) ||
          !data[key])
        return key;
      else return data[key];
    }
  }
])
```

A kezdőoldalon található egy carousel, ami lapozható képeket tartalmaz. Ng-repeat ciklussal lettek a képek ismételve elhelyezve.

```
<!-- Content -->
<div class="row mt-5">
  <hr>
  <h4 class="">{{'homenews' | translate:lang.data}}</h4>
</div>
<!-- Rental process -->
<ul class="row list-unstyled text-center mt-4">
  <li class="col-12 col-lg-6 my-2"
    ng-repeat="x in [].constructor(4) track by $index">
    <div class="cardbox p-2 login_container my-3 mx-3">
      <div class="row row-cols-1 row-col-md-2 g-4">
        <div class="col-6">
          <div class="position-relative mt-2">
            
          </div>
        </div>
      </div>
    </div>
  </li>
</ul>
```

```
<!-- Carousel images -->
<div class="carousel-inner">
  <div class="carousel-item"
    ng-class="{ 'active': $index === 0}"
    ng-repeat="x in carousel track by $index">
    
    class="d-block w-100"
    alt="img">
  </div>
</div>
```

Ezek alatt rövid hírek találhatóak szintén a fentebb említett eljárással készítve.

A képzések menüponton belül mySQL adatbázisba lévő képzések kerültek, PHP által meghívva, amit egy javascript controller vezérel Ajax kéréssel.

```
// Courses controller
.controller('coursesController', [
  '$scope',
  'http',
  function($scope, http) {

    // Http request
    http.request('./php/courses.php')
      .then(response => {
        $scope.courses = response;
        $scope.$applyAsync();
      })
      .catch(e => console.log(e));
  }
]);
```

```
// Connect to database
$db = new Database();



















// Set query
$query = "SELECT * FROM courses;";

// Execute query with argument
$result = $db->execute($query);

foreach($result as $i => $row) {

  // Set query
  $query = "SELECT *
            FROM `courses_subjects`
            INNER JOIN `subjects`
            ON `subjects`.`subject_id` = `courses_subjects`.`subject_id`
            WHERE `course_id` = :id";

  // Execute query with argument
  $result[$i]['subjects'] = $db->execute($query, array("id" => $row['course_id']));
}
```

				course_id	course_name	description
<input type="checkbox"/>	 Módosítás	 Másolás	 Törölés	1	courses1	courses1_description
<input type="checkbox"/>	 Módosítás	 Másolás	 Törölés	2	courses2	courses2_description
<input type="checkbox"/>	 Módosítás	 Másolás	 Törölés	3	courses3	courses3_description
<input type="checkbox"/>	 Módosítás	 Másolás	 Törölés	4	courses4	courses4_description
<input type="checkbox"/>	 Módosítás	 Másolás	 Törölés	5	courses5	courses5_description
<input type="checkbox"/>	 Módosítás	 Másolás	 Törölés	6	courses6	courses6_description

Ezekre a képzésekre kattintva (amit a képzés képét kijelölve megjelenő gombra kattintva tudunk előhozni) előjön egy részletes leírás.

```
<!-- Row -->
<div class="row mt-3">
  <div ng-repeat="x in course.subjects">
    <h5 class="mt-3">{{x.subject_name | translate:lang.data}}</h5>
    <p>{{x.description | translate:lang.data}}</p>

    <div class="mt-2"
      | ng-repeat="y in x.teachers">
      <h6>{{'instructor' | translate:lang.data}}:
      | {{y.surname}} {{y.first_name}}
    </h6>
    </div>
  </div>
</div>
```

```
<div class="text-center mb-1">
  <button class="btn-courses btn btn-sm btn-primary px-4 shadow-sm"
    | ng-class="{ 'disabled': !user.id}"
    | data-bs-toggle="modal"
    | data-bs-target="#my-modal-1">
    <span class="text-capitalize">
      | {{'apply' | translate:lang.data}}
    </span>
  </button>
</div>

<div class="text-center mb-5">
  <span class="fst-italic fs-6"
    | ng-class="{ 'd-none': user.id}">
    | {{'needlogin' | translate:lang.data}}
  </span>
</div>
```

Itt a lap alján található jelentkezés gomb csakis bejelentkezés után látható. A bejelentkezést az ng-class ellenőrzi, amennyiben nincs bejelentkezve a felhasználó (user.id=0), a gomb disabled, illetve egy szöveg megjelenik alatta, hogy bejelentkezés szükséges.

Jelentkezés

Jelentkezés

Bejelentkezés szükséges!

Jelentkezéskor az adatbázisba kerül a felhasználó id-ja, a képzés id, amire jelentkezik, a képzés kezdési ideje, illetve a tanár.

```
// Get arguments
$args = Util::getArgs();

// Connect to database
$db = new Database();

// Set query
$query = "INSERT INTO `students`
        (`course_id`, `beg`, `user_id`)
        VALUES (:courseId, :beg, :userId)";

// Execute query with argument
$result = $db->execute($query, $args);

// Close connection
$db = null;

// Set response
Util::setResponse($result);
```

Jelentkezés



Smart home rendszerek tervezése és üzemeltetése szak

Kérlek válassz az alábbi kezdési időpontok közül:

2024-02-05



☒ A jelentkezéssel megértem és elfogadom az iskola [adatvédelmi szabályzatát](#).

Jelentkezés

Mégsem

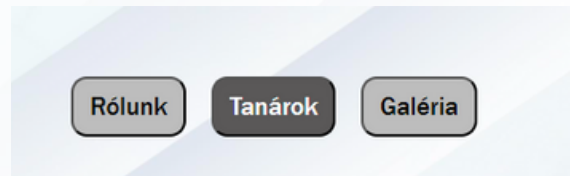
Ezek az adatbázisok össze vannak kapcsolva.

A közösség oldalon 3 tab gomb található, szintén Bootstrap-pel és CSS-sel megformázva, a tanárok tab alatt adatbázisba feltöltött tanárok adatai találhatóak, A tanár nevére kattintva egy modal segítségével felugró ablakban megjelenik a tanár képe, illetve egy rövid bemutatkozás.

```
// Community controller
.controller('communityController', [
  '$scope',
  'http',
  function($scope, http) {

    $scope.galleryImgIndex = -1;
    $scope.galleryClicked = (event) => {
      let element = event.currentTarget;
      index = element.getAttribute('data-index');
      $scope.galleryImgIndex = parseInt(index);
    };
    $scope.teacherData = (event) => {
      let element = event.currentTarget;
      $scope.teacherImg = element.dataset.img;
      $scope.teacherContent = element.dataset.content;
      $scope.teacherFull_name = element.dataset.full_name;
      $scope.$applyAsync();
    }
  }
]);

// Http request
http.request('./php/teachers.php')
.then(response => {
  $scope.teachers = response;
  $scope.$applyAsync();
})
.catch(e => console.log(e));
}
```



```
<!-- Tab2 -->
<li class="nav-item nav-item-pills mx-2"
    role="presentation">

  <!-- Button -->
  <button id="tab-2-tab"
    class="comm_btn btn-outline-secondary ms-2 mb-2
      btn-pills-tab text-capitalize"
    data-bs-toggle="pill"
    data-bs-target="#tab-2"
    role="tab"
    type="button"
    aria-controls="tab-2"
    aria-selected="false">
    {{'teacher' | translate:lang.data}}
  </button>
</li>
```

```
<!-- Modal // Teacher description-->
<div id="my-modal-1"
  class="modal fade"
  data-bs-dismiss="modal"
  data-bs-backdrop="static"
  ata-bs-keyboard="true"
  tabindex="-1"
  aria-labelledby="staticBackdropLabel"
  aria-hidden="true">

  <div class="modal-dialog modal-dialog-centered modal-dialog-scrollable">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title"
          id="staticBackdropLabel">
          {{teacherFull_name}}
        </h5>

        <button type="button"
          class="btn-close me-2"
          data-bs-dismiss="modal"
          aria-label="Close">
        </button>
      </div>

      <div class="modal-body">
        
        <p>{{teacherContent | translate:lang.data}}</p>
      </div>
    </div>
  </div>
```


A galéria fül alatt
képek találhatóak
amik tovább
nagyíthatók az
előbb említett
modal
használatával,
amelyek ng-
repeat ciklussal
vannak
ismételtetve.

```
<!-- Tab3 panel // gallery-->
<div id="tab-3"
  class="tab-pane fade p-3 text-center"
  role="tabpanel"
  aria-labelledby="tab-3-tab">

  <div class="row justify-content-center">
    <div class="col-lg-4 col-md-12 mb-4 mb-lg-0">

      <div ng-repeat="x in [].constructor(16) track by $index"
        data-index="{{ $index+1 }}"
        ng-click="galleryClicked($event)">
        <a>
          
          </a>
        </div>
      </div>
```

A kapcsolat menüpontban egy form található melyben 4 input mező
kapott helyet, amik a community controllert használva php segítségével
email küldésre alkalmasak.

```

<?php

use Util\Util as Util;
use Language\Language as Language;
use PHPMailer\Email as Email;

require_once('.../common/php/environment.php');

$args = Util::getArgs();

// Set language
$lang = new Language($args['lang']);

// Translate error messages
$errorMsg = $lang->translate(Email::$errorMessages);

$langData = $lang->translate(array(
    "{{confirmation}}" => "confirmation",
    "{{returnText}}"   => "returnText",
    "{{yourMessage}}"  => "yourMessage",
    "{{school}}"       => "school"
));

// Set constants data
$constants = array(
    "{{lang_id}}"      => $args['lang']['id'],
    "{{current_date}}" => date('Y/m/d H:i:s'),
    "{{current_year}}" => date("Y"),
    "{{message}}"      => $args['params']['message'],
    "{{email}}"        => $args['params']['email'],
    "{{fromName}}"     => $args['params']['name']
);

```

Név...

Tárgy...

Az Ön e-mail címe...

Üzenet...

Küldés!

A küldés gombra kattintva kap egy emailt az iskola, melyben lista szerint le van írva a 4 input mező tartalma, a felhasználó pedig egy megerősítő emailt kap az oldalon előre beállított nyelven, amely szintén tartalmazza az üzenete tartalmát is.

PHPMailer gondoskodik az üzenetküldésről, amely kiemeli az argumentumokat a controlleren keresztül, amivel aztán dolgozni tud a "rendszer" és 2 html oldal kerül küldésre.

```
// Az email tartalma

$phpMailerSchool->FromName = $args['params']['name'];
$phpMailerUser->Subject = $args['params']['subject'];
$phpMailerSchool->Subject = $args['params']['subject'];
$phpMailerUser->Body      = $phpMailerUser->getDocument();
$phpMailerSchool->Body      = $phpMailerSchool->getDocument();
$phpMailerUser->addAddress($args['params']['email']);
$phpMailerSchool->addAddress('jakabhunorszki@gmail.com');
$phpMailerUser->isHTML(true);

// Send email
$phpMailerUser->send();
$phpMailerSchool->send();
```

```
// Visszaigazolás az email elküldéséről

$phpMailerUser->setDocument(array(
    'fileName'      => "contactUser.html",
    'subFolder'     => 'email'
), $constants, $langData);

// Az iskola részére küldött email

$phpMailerSchool->setDocument(array(
    'fileName'      => "contactSchool.html",
    'subFolder'     => 'email'
), $constants, $langData);
```

```

<!DOCTYPE html>
<html lang="{{lang_id}}">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>{{contact}}</title>
</head>

<body>
  <div style="text-align: center; background-color: ■rgb(223, 228, 227);
    width: fit-content; height: fit-content;">
    <h1><u>{{confirmation}}</u></h1>
    <h1>{{greetings}} {{fromName}}!</h1>
    <h2>{{returnText}}</h2>
    <h5>{{yourMessage}}</h5>
    <p><i>{{message}}</i></p>
    <p style="font-size: smaller;"><i><u>{{school}}</u></i></p>
    <p style="font-size: x-small;"><i>{{current_date}}</i></p>
  </div>
</body>

</html>

```

Az érdeklődőnek
küldendő üzenet kódja.

Az iskolának küldendő
üzenet kódja.

```

<!DOCTYPE html>
<html lang="{{lang_id}}">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>{{contact}}</title>
</head>

<body>
  <h1>Beérkezett üzenet</h1>
  <div style="background-color: ■rgb(223, 228, 227); width: 400px;
    height: fit-content;">
    <ul>
      <li><u>Név:</u> {{fromName}}</li>
      <li><u>E-mail cím:</u> {{email}}</li>
      <li><u>Üzenet:</u> {{message}}</li>
      <li><u>Üzenetküldés ideje:</u> {{current_date}}</li>
    </ul>
  </div>
</body>
</html>

```

A bejelentkezés gombra kattintva meghívásra kerül a login.html oldal, amely egy form, ng-pattern által ellenőrzött e-mail cím szükséges továbbá a jelszónál is regex használatával van megoldva, hogy a tartalma kis- és nagybetű, illetve számok legyenek.

Bejelentkezés

★ ✉ Email cím:

gera.denes@keri.mako.hu

✕

✓

★ 🔑 Jelszó:

jelszó

6-20 karakter (kis-nagy betűt, és számot tartalmaz) .

☐ 👁 jelszót megjelenít

★ 🏷 Kód:

uZs7v

🏠 kód

↻ Frissít

```
<!-- Input email -->
<div class="col-11 col-md-7 position-relative">
  <input id="email"
    name="email"
    type="email"
    ng-attr-placeholder="{{('email_address' | translate:lang.data)}}"
    class="form-control form-control-sm input-email"
    spellcheck="false"
    autocomplete="off"
    required
    maxlength="50"
    ng-model="model.email"
    ng-init="model.email=user.email"
    ng-trim="false"
    ng-disable-space
    ng-pattern="/^([a-z0-9_.-]+)\@((([a-z0-9-])+\.)+([a-z0-9]{2,4})+$)/"
    ng-clear-icon
    ng-check-mark>
```


Ugyanígy a regisztráció is működik, checkmarkok jelölik, ha megfelelő a formátum, illetve van tartalma az inputnak. Ezek az adatok az INSERT sql paranccsal lesznek beépítve a user táblánkba.

```
$fields = array_filter(array(
    "type" => null,
    "name" => null,
    "prefix_name" => null,
    "first_name" => null,
    "middle_name" => null,
    "last_name" => null,
    "suffix_name" => null,
    "nick_name" => null,
    "born" => null,
    "gender" => null,
    "img" => null,
    "img_type" => null,
    "country" => null,
    "country_code" => null,
    "phone" => null,
    "city" => null,
    "postcode" => null,
    "address" => null,
    "email" => null,
    "password" => null,
    "created" => null,
    "email_verification_code" => null
), function($key) use($userFields) {
    return array_key_exists($key, $userFields);
}, ARRAY_FILTER_USE_KEY);

// Set query
$query = "INSERT INTO `user`
    (`" . implode("`", array_keys($fields)) . "`)
VALUES";
```

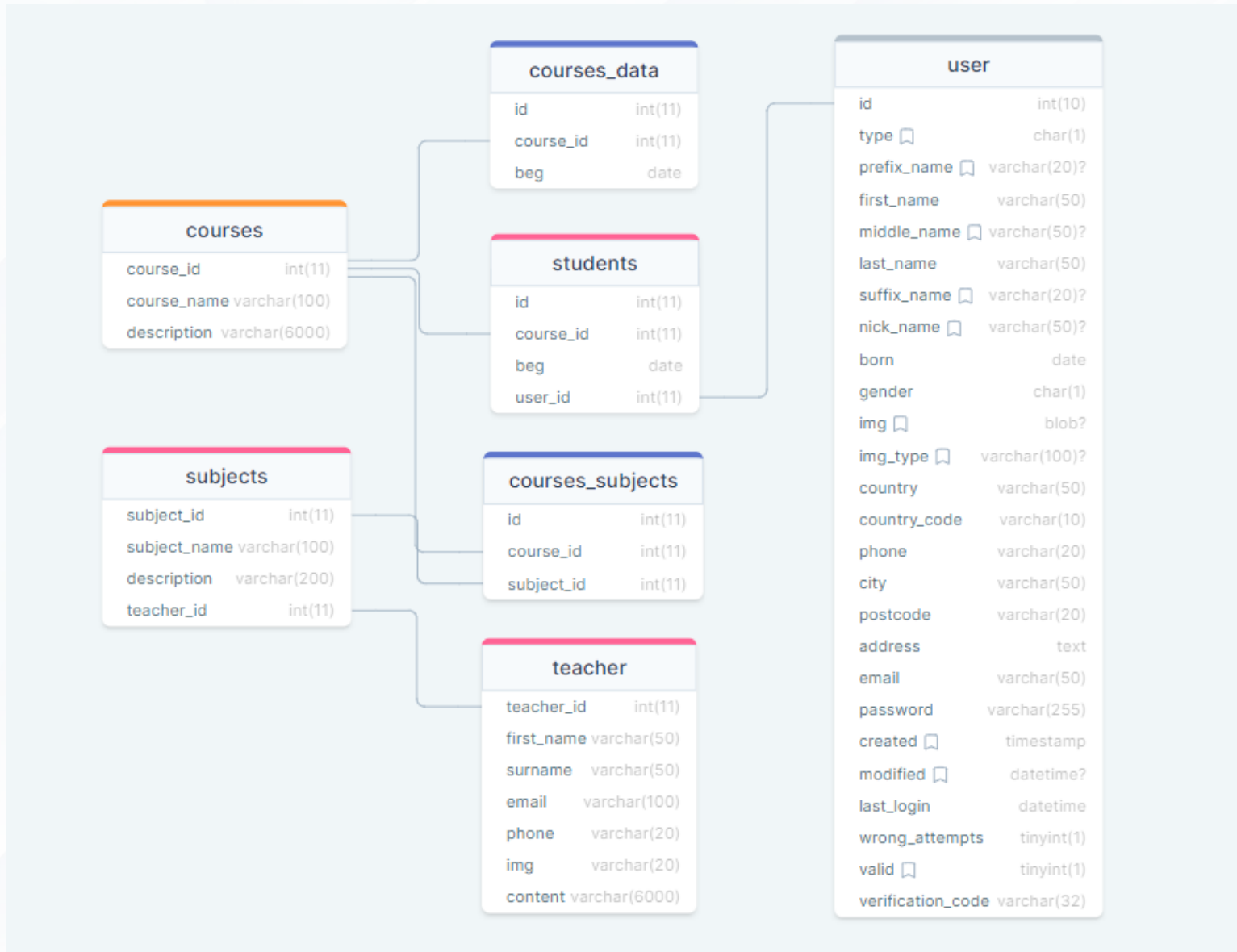
The image shows a registration form with the following fields and validation status:

- Város:** város (no validation icon)
- Irányítószám:** irányítószám (no validation icon, with a note: "4-10 számjegy, és nem kezdődhet nullával.")
- Cím:** cím (no validation icon)
- Email cím:** gera.denes@keri.mako.hu (validation icon: ✕)
- Email cím megerősítés:** gera.denes@keri.mako.hu (validation icon: ✓)
- Jelszó:** jelszó (validation icon: ✕, with a note: "6-20 karakter (kis-nagy betűt, és számot tartalmaz).")
- Jelszó megerősítés:** jelszó megerősítés (validation icon: ✕)

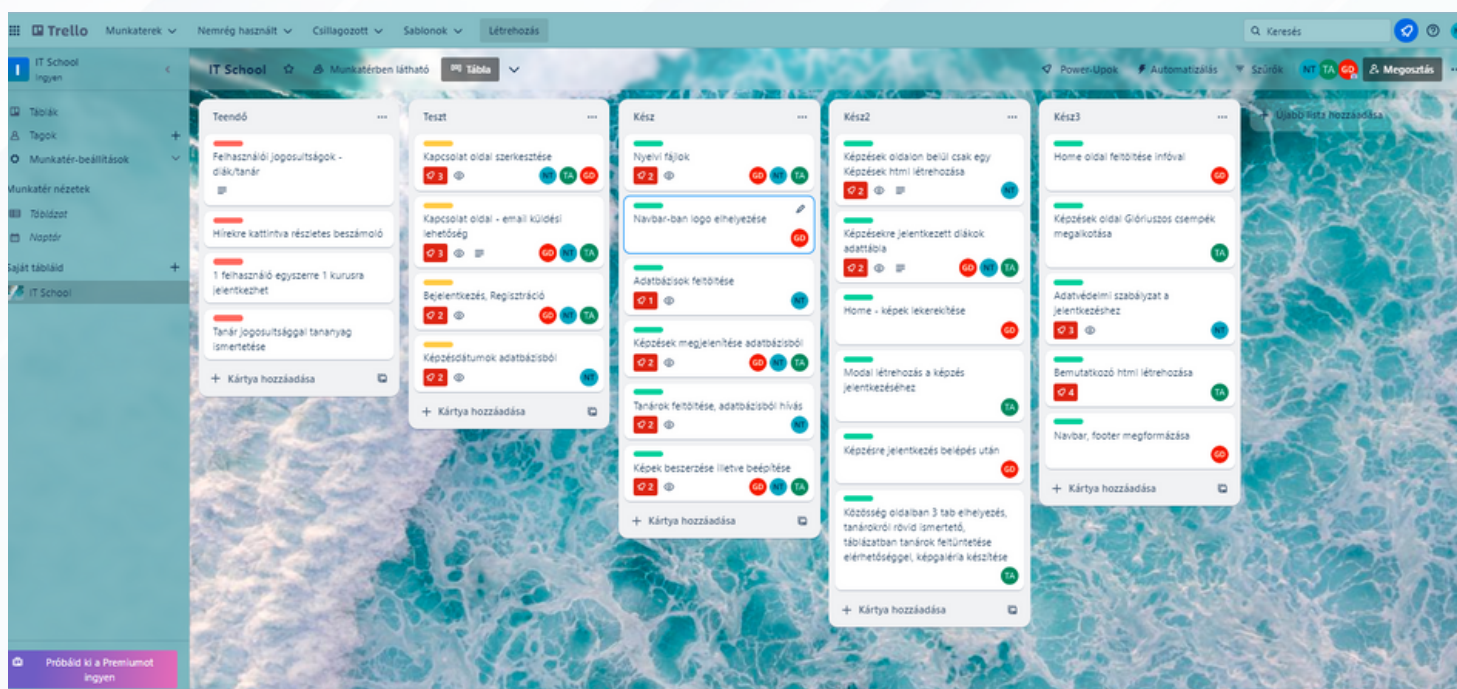
There is also a checkbox labeled "jelszót megjelenít" which is currently unchecked.

Adatok módosítására is lehetőség van, a fent említett eljárások alkalmazásával.

Az oldal által használt adatbázis 7 táblát tartalmaz.



A csapatmunka során az elvégzendő feladatok kiválasztását, folyamatos nyomonkövetését és elvégzésének visszajelzését a Trelloval oldottuk meg.



A GitHub-ot használtuk verziókezeléshez.

Gera Dénes Takács Ádám Németh Tibor