

Веб-Апликација - ФИНКИ Форум и Студиски Водич

Теза по Напреден Веб Дизајн при
Факултет за Информатички Науки и Компјутерско Инженерство

2024

2024 Даниел Стојановски

Апстрактција

Системот ФИНКИ Форум и Студиски Водич е веб-апликација наменета за сите досегашни и идни студенти на Факултетот за Информатички Науки и Компјутерско Инженерство, која нуди два модули, каде што едниот модул е Форум кој корисниците може да го користат за споделување искуства и здобивање на информации, додека другиот модул, Водич, е општо наменет за информирање за студиите на Факултетот за Информатички Науки и Компјутерско Инженерство и професиите од секторот на интернет технологиите.

Содржина

Вовед	4
Модул Форум	5
Имплементација	10
Модул Водич	13
Имплементација	18
Најава и Регистрација	21
Имплементација	22

Вовед

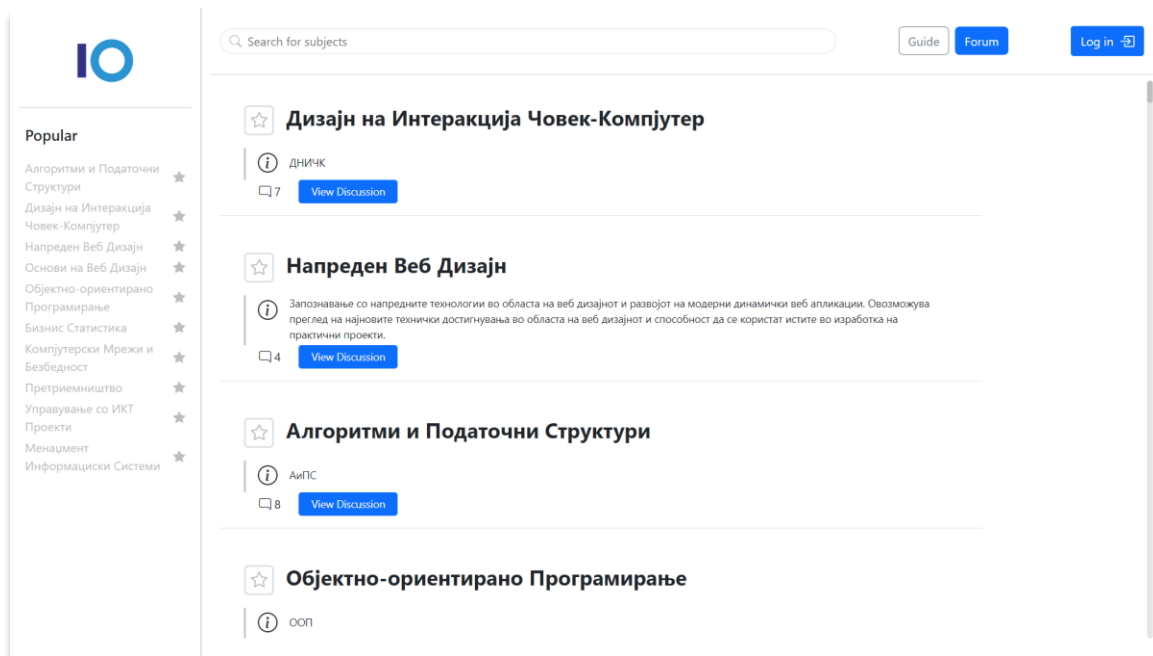
Веб-апликацијата ФИНКИ Форум и Студиски Водич е МРА (multiple-page application) изработена со помош на повеќе алатки, од кои, Vue.js се користи како framework, во придружба на Express.js и Node.js како API сервис за поставување на серверот, поврзано со податочна база изработена во рамки на PostgreSQL, и Bootstrap за визуелна репрезентација.

Модул Форум

Модулот Форум е наменет да биде збирно место за интеракција, споделување на информации, прашања и образложенија за сите корисници на веб-апликацијата.

Почетниот приказ на Форум модулот е составен од листа на дискусии на тема на предмети кои се изучуваат во рамки на студиите на Факултетот, странично мени на кое стои листа од најпопуларните дискусии, и навигациско мени на врвот на приказот на кое стои пребарувач, навигациски копчиња меѓу модулите и копче за најава.

На Слика 1 е прикажан почетниот приказ на Форум модулот:



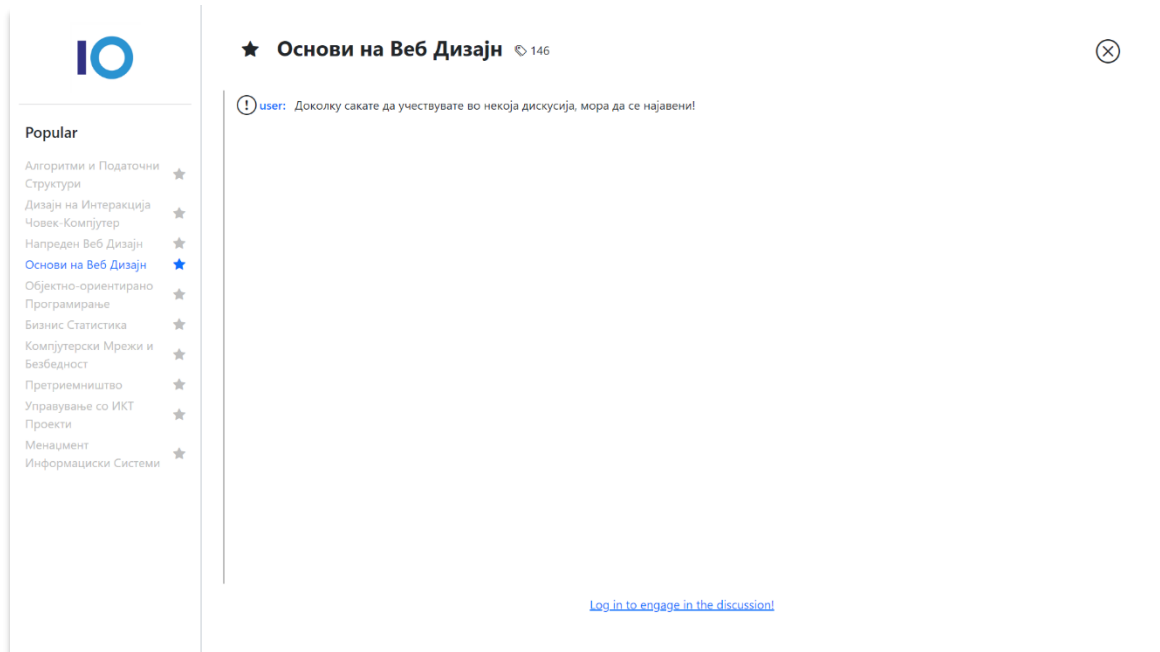
Слика 1. Почетен приказ на Форум модулот, без најавен корисник.

Корисникот, во случајот ненајавен, има можност да посети дискусија преку копчето кое се наоѓа во рамки на секој поединечен елемент од листата.

При притискање на копчето, или некоја дискусија во рамки на страничното мени од најпопуларни дискусии, корисникот е пренасочен на следниот приказ од Форум модулот. Овој приказ ја прикажува содржината на целата дискусија, поточно структурата на коментарите поставени во рамките на дискусијата.

Бидејќи корисникот е сеуште ненајавен, тој нема можност за интеракција со содржината на дискусијата освен пристап до истата.

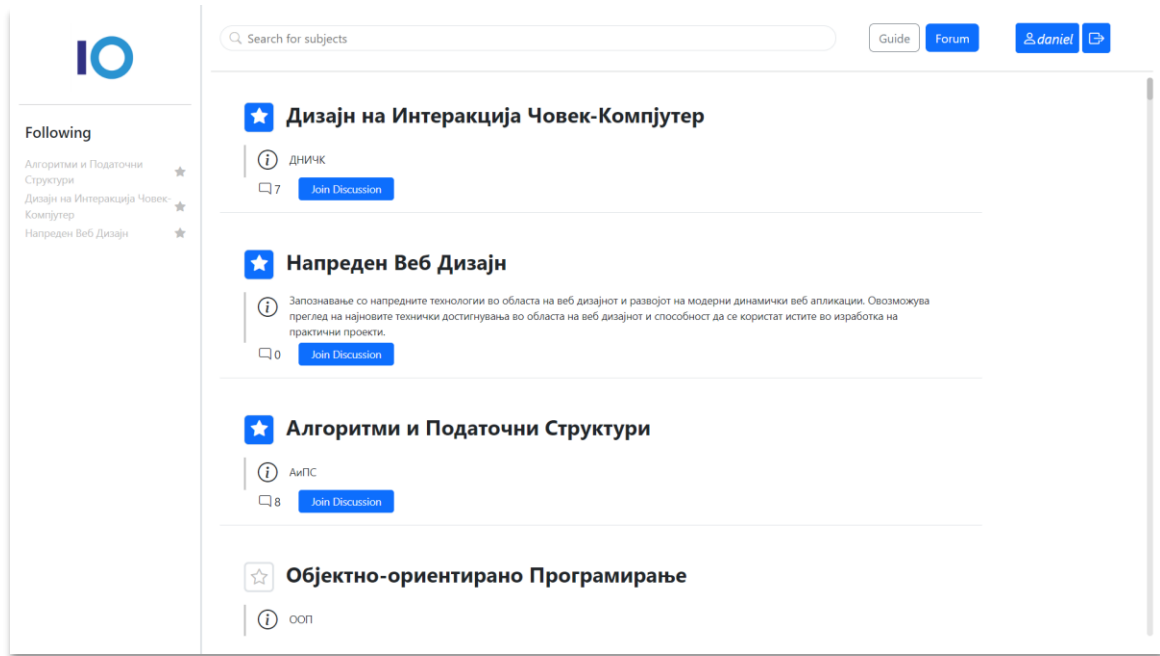
На Слика 2 е прикажан приказот на содржината на една дискусија:



Слика 2. Приказ на содржина на дискусија на Форум модулот, без најавен корисник.

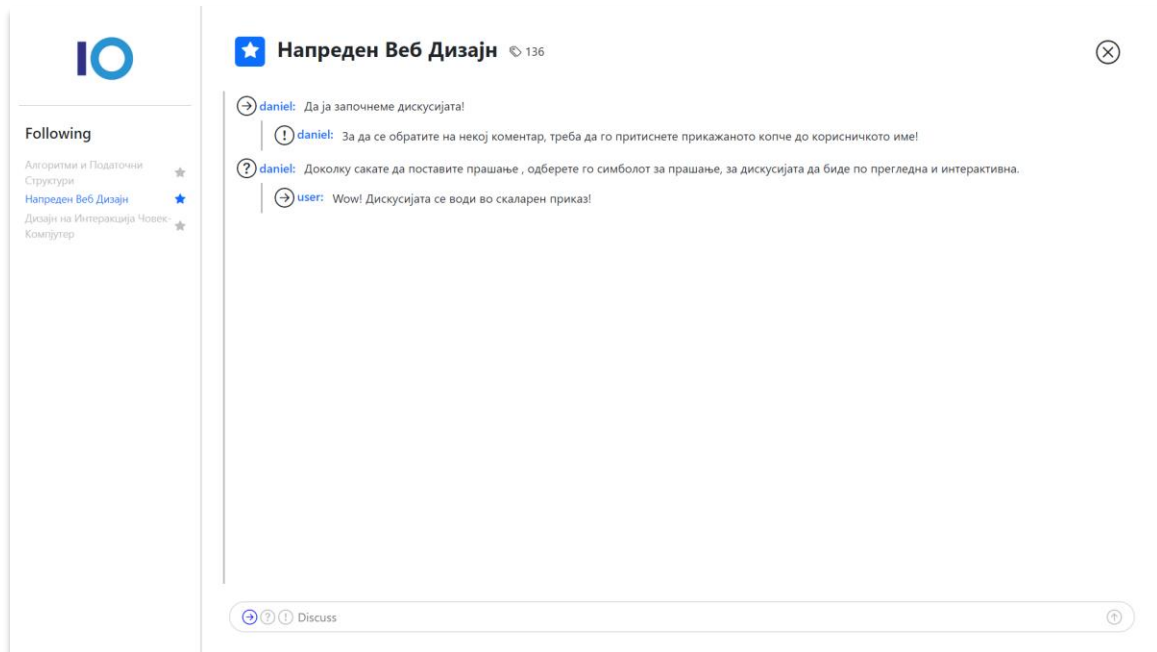
Штом корисникот се најави, тој има можност да заследува дискусии и можност за интеракција со содржината на дискусијата.

Приказот на веб-апликацијата откако корисникот ќе се најави е прикажан на следната слика, Слика 3.1:



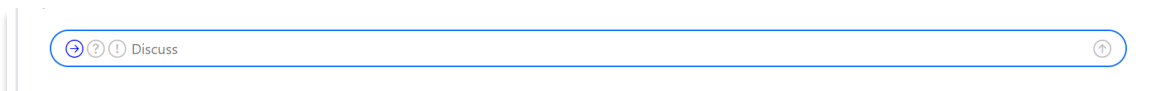
Слика 3.1. Почетен приказ на Форум модулот, кај најавен корисник.

Приказот на веб-апликацијата откако корисникот ќе одбере да се приклучи на некоја дискусија при притискање на копчето за придружба е прикажан на следната слика, Слика 3.2:

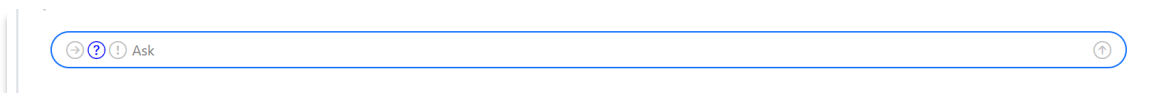


Слика 3.2 Приказ на содржина на дискусија на Форум модулот, кај најавен корисник.

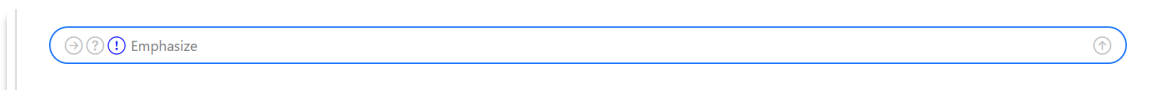
Корисникот добива можност своите коментари да ги означи како прашања или пак образложенија. Изгледот на полето за внес на коментари, во секоја од фазите, е прикажано во следната серија слики, со назив Слика 4:



Слика 4.1. Изглед на полето за внес на коментари, при нормален коментар.



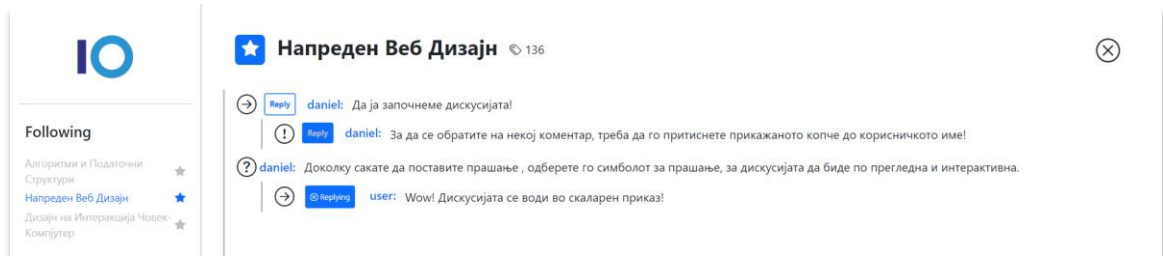
Слика 4.2. Изглед на полето за внес на коментари, при поставување на прашање.



Слика 4.3. Изглед на полето за внес на коментари, при доделба на образложение.

Понатаму, корисникот има и можност да се приклучи во разговор со останатите корисници, преку возвраќање на нивниот коментар со под-коментар (скалесто распоредување на коментарите).

На Слика 5 се прикажани сите фази на копчето за возвраќање на коментар:



Слика 5. Изглед на сите три фази за копчето за возвраќање на коментар.

Доколку се навратиме на Слика 5. може да ги увидиме трите фази на копчето за возвраќање на коментар. Да ги образложиме карактеристиките на овие три фази:

- Општа состојба – При преминување врз еден коментар, се појавува копчето за возвраќање на коментари, кое во оваа фаза е неисполнето и само обележано со сина боја.
- Покриена состојба – При покривање врз копчето за возвраќање на коментар, копчето преминува во следната фаза и го менува својот изглед и се исполнува со сина боја.
- Избрана состојба – При избирање на копчето за возвраќање на коментар, тоа преминува во последната фаза која го прикажува копчето со придружба на

икс-симбол и го менува текстот т.е глаголот во несвршено време. Тој го задржува својот приказ се додека не се избере повторно, откако копчето ќе биде повторно избрано тоа ја прекинува неговата функција и корисникот е ослободен од возвраќање на избраниот коментар.

Имплементација

Листата на дискусии е составена од повеќе `<Discussion>` компоненти чиј код изгледа вака:

```
<div v-for="subject in subjects" :key="subject.id" class="col-md-10">
  <div class="card border-0 border-bottom rounded-0">
    <div class="card-body mx-3">

// Копче за заследување и наслов на Дискусијата
    <div class="d-flex align-items-center">
      <button @click="toggleFavorite(subject)"
        :class="{ 'border-3': !subject.isFavorite,
          'btn-primary': subject.isFavorite }"
        class="btn border d-flex justify-content-center pt-0"
        style="width: 40px; height:40px">
        <i :class="{ 'bi-star': !subject.isFavorite,
          'text-gray': !subject.isFavorite, }"
          class="bi bi-star-fill m-auto"
          style="font-size: 22px;"></i>
        </button>
      <h3 class="p-3 card-title fw-bold">
        {{ subject.name }}
      </h3>
    </div>
    ...
  </div>
</div>
```

```

...
// Информации за Дискусијата
<div class="card-text body-line-left d-flex pb-1">
  <i class="bi bi-info-circle"></i>
  <p class="mx-3 my-auto text-wrap" style="width: 100%;">
    {{ subject.info }}
  </p>
</div>

// Број на коментари во Дискусијата
<div class="d-flex mx-1">
  <span class="px-2 mx-2 my-auto">
    <i class="bi bi-chat-right"></i>
    {{ subject.comments_count }}
  </span>

// Копче за пренасочување
  <router-link :to="'/board/'+subject.id">
    <button class="px-3 mx-2 my-auto btn btn-primary btn-sm">
      <span v-show="isLoggedIn">Join Discussion</span>
      <span v-show="!isLoggedIn">View Discussion</span>
    </button>
  </router-link>
</div>
</div>
</div>
</div>

```

Копчето за заследување и иконата во него го менуваат својот изглед зависно од променливата „isFavorite“ која е особина на секој предмет во листата на предмети од базата на податоци. Променливата и овозможува на дискусијата да биде доделена во листата на заследени дискусии. Зависно од променливата, копчето е исполнето во сина боја или е празно.

Копчето за пренасочување пак, го менува својот натпис зависно од тоа дали корисникот е најавен или не.

Начинот на заследување на дискусија е изработен со помош на localStorage, прикажано во метод:

```

async toggleFavorite(subject) {
  try {
    if (subject.isFavorite) {
      await this.$axios.delete(`/api/favorites/${subject.id}`);
      localStorage.setItem(`favorite_${subject.id}`, 'false');
    } else {
      await this.$axios.post('/api/favorites', {
        id: subject.id,
        name: subject.name
      });
      localStorage.setItem(`favorite_${subject.id}`, 'true');
    }
    subject.isFavorite = !subject.isFavorite;
  }
}

```

За интеракција во рамки на дискусијата, корисникот ја користи `<InputField>` компонентата, чиј код изгледа на следниот начин:

```

<div class="d-flex mx-4 input-group-text p-0 mt-2 px-3 p-1
  btn-outline-gray border-2 rounded-5 bg-white input-container"
  style="color: #bebebe" v-if="isLoggedIn">
  //Редица од ознаки за коментари
  <i class="bi bi-arrow-right-circle"
    :class="{ 'text-primary': selectedIcon === 'discuss' }"
    @click="selectIcon('discuss')"></i>
  <i class="bi bi-question-circle mx-1"
    :class="{ 'text-primary': selectedIcon === 'ask' }"
    @click="selectIcon('ask')"></i>
  <i class="bi bi-exclamation-circle"
    :class="{ 'text-primary': selectedIcon === 'emphasize' }"
    @click="selectIcon('emphasize')"></i>
  //Поле за внес на коментар
  <input class="mx-1" style="width: 100%;" type="text"
    v-model="commentText"
    @keydown.enter="postComment(selectedCommentId,
      selectedCommentDepth)"
    :placeholder="placeholderText" aria-label="Comment">
  //Ознака за испраќање на порака
  <i class="bi bi-arrow-up-circle"
    @click="postComment(selectedCommentId,
      selectedCommentDepth)"></i>
</div>

```

Доколку корисникот е ненајавен, кодот на компонентата е следниот:

```

<div class="d-flex" v-if="!isLoggedIn">
  <router-link class="m-auto" to="/login">
    Log in to engage in the discussion!
  </router-link>
</div>

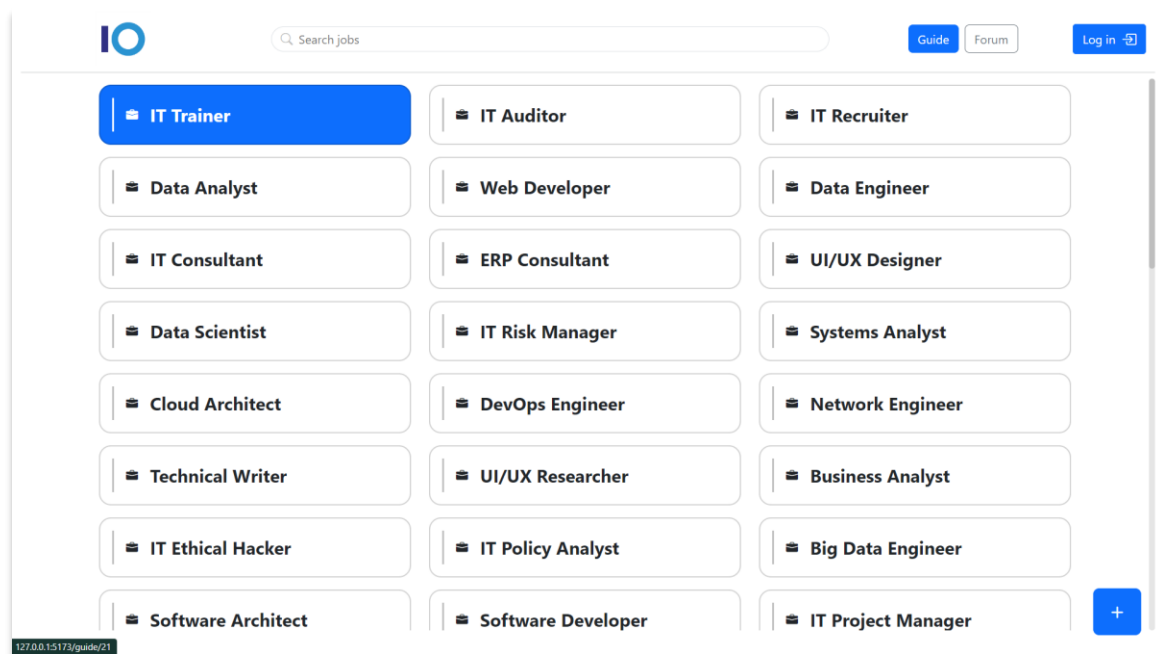
```

Модул Водич

Модулот Водич е наменет да биде информативно место за сите корисници на веб-апликацијата, особено за корисници кои имаат страст да бидат студенти на факултетот.

Почетниот приказ на Водич модулот е составен од листа од картички од професии од секторот на интернет технологии и навигациско мени на врвот на приказот на кое стои пребарувач, навигациски копчиња меѓу модулите и копче за најава.

На Слика 6 е прикажан почетниот приказ на Водич модулот, каде е прикажа листа од картички со професии:



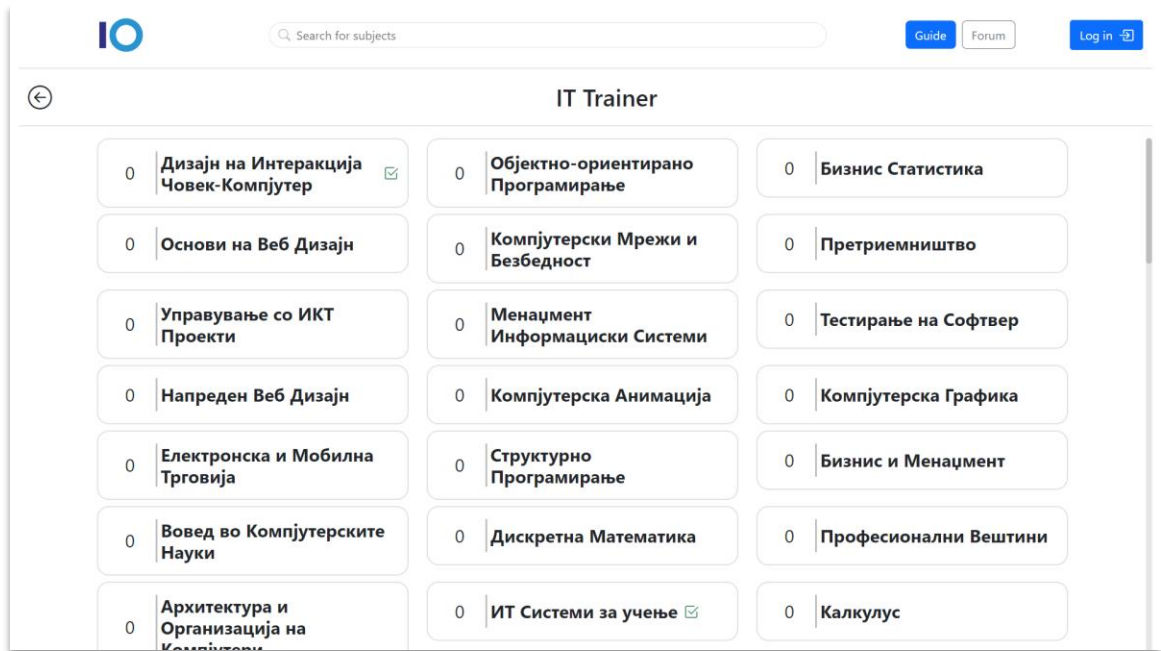
Слика 6. Почетен приказ на Водич модулот, без најавен корисник.

Корисникот, во случајот ненајавен, има можност да отвори картичка со професија преку притискање на истата, со што е пренасочен на подетален приказ.

Откако корисникот ќе избере картичка со професија, преминува на следниот приказ од Водич модулот. Овој приказ овозможува корисникот да ја прегледа релацијата на професијата со секој од студиските предмети.

Новите картички со предмети прикажуваат број на гласови и ознака за значајност. Бројот на гласови е акумулација од гласовите на сите најавени корисници кои оствариле глас за специфичниот предмет во релација со професијата. Додека ознаката за значајност е поставена системски како обележје за коректна релација на предметот со професијата.

Изгледот на следниот приказ на Водич модулот откако ќе корисникот ќе одбере картичка со професија е прикажан на следната слика, Слика 7.:



Слика 7. Приказ на релации Професија-Предмет во Водич модулот, без најавен корисник.

Корисникот доколку е најавен има можност да додаде професија по желба и на неа да и назначи задолжителни релативни предмети од листата на студиски предмети од факултетот.

Приказот на оваа функционалност во рамки на Водич модулот е прикажан на Слика 8:

IO Search jobs Guide Forum user

Add Job

Job name

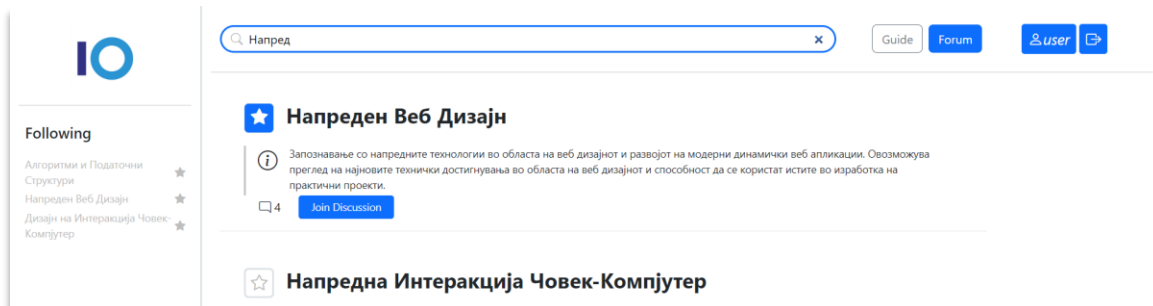
<input type="checkbox"/> Дизајн на Интеракција Човек-Компјутер	<input checked="" type="checkbox"/> Објектно-ориентирано Програмирање	<input type="checkbox"/> Бизнес Статистика
<input type="checkbox"/> Основни на Веб Дизајн	<input type="checkbox"/> Компјутерски Мрежи и Безбедност	<input type="checkbox"/> Претприемништво
<input type="checkbox"/> Управување со ИКТ Проекти	<input type="checkbox"/> Менаџмент Информациски Системи	<input type="checkbox"/> Тестирање на Софтвер
<input checked="" type="checkbox"/> Напреден Веб Дизајн	<input checked="" type="checkbox"/> Компјутерска Анимација	<input checked="" type="checkbox"/> Компјутерска Графика
<input type="checkbox"/> Електронска и Мобилна Трговија	<input type="checkbox"/> Структурно Програмирање	<input type="checkbox"/> Бизнес и Менаџмент
<input type="checkbox"/> Вовед во Компјутерските Математики	<input type="checkbox"/> Дискретна Математика	<input type="checkbox"/> Професионални Вештини

Finish

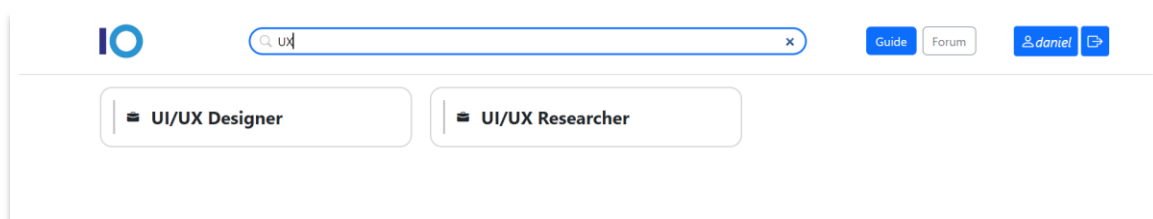
Слика 8. Приказ на функцијата за додавање нова професија, при најавен корисник.

На Слика 8, може да се забележи изгледот на картичка за селекција, од која е изградена листата на предмети. Секоја инстанца од компонентата поседува сопствена променлива ознака која ја обележува селекцијата на предметот. и поле за називот на новата професија, и копче за верификација на додавање на професијата во системот на Водич модулот.

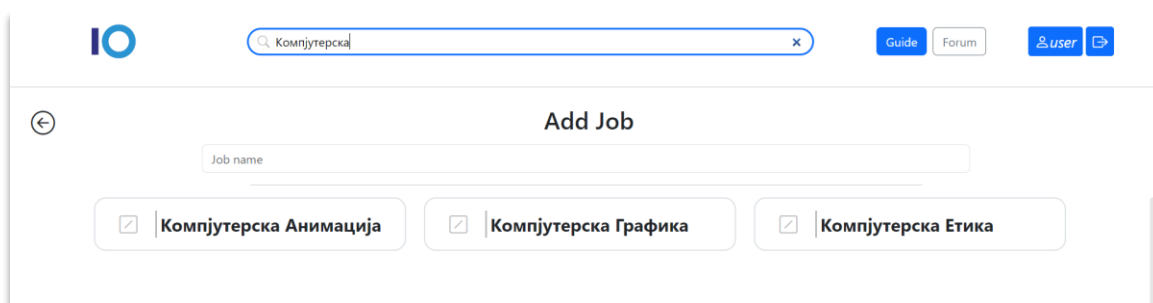
Во рамките на двата модули, полето за пребарување филтрира низ карактеристичната листа на елементи зависно од модулот. Тоа е прикажано на следната серија слики:



Слика 9.1. Пребарување на листата на дискусии



Слика 9.2. Пребарување на листата на професии



Слика 9.3. Пребарување на листата на предмети, при додавање на професија

Имплементација

Листата на Картички од професии е составена од повеќе `<JobCard>` компоненти чиј код изгледа вака:

```
<div v-for="role in roles" :key="role.id" class="col-md-4">
  <router-link :to="'/guide/'+role.id" style="text-decoration: none;">
    <div class="card card-hover my-2 rounded-4 border-2">
      <div class="card-body btn border-0">
        <div class="d-flex align-items-center body-line-left">
          <i class="bi bi-briefcase-fill"></i>
          <h4 class="px-3 py-2 my-auto fw-bolder">
            {{ role.role_name }}
          </h4>
        </div>
      </div>
    </div>
  </router-link>
</div>
```

Се користи Vue тагот `<router-link>` за да се овозможи пренасочување помеѓу повеќето прикази на Водич модулот, всушност кон листата на релациски предмети.

```
async fetchRoles() {
  try {
    const response = await this.$axios.get('/api/roles');
    this.roles = response.data;
  } catch (error) {
    console.error('Error fetching roles:', error);
  }
}
```

Како метод за повлекување на податоците од базата на податоци користи Axios барање, кое се сведува на Express барање од Postgres базата означена како „pool“.

```
await pool.query('SELECT * FROM it_roles ORDER BY LENGTH(role_name) ;')
```

Листата од Картички од предмети во релација со професијата е составена од повеќе

<SubjectCard> компоненти чиј кон изгледа вака:

```
<div v-for="subject in subjects" :key="subject.id" class="col-md-4">
  <div class="d-flex py-3 px-2 rounded-4 border border-2">
    <div class="mx-4 m-auto justify-content-center align-items-center">
      <i class="bi bi-arrow-up-circle" v-show="isLoggedIn"
        @click="vote(subject.id, 1)"></i>
      <div class="d-flex">
        <p class="p-1 my-auto">{{ getVoteCount(subject.id) }}</p>
      </div>
      <i class="bi bi-arrow-down-circle" v-show="isLoggedIn"
        @click="vote(subject.id, -1)"></i>
    </div>
    <div class="card-body body-line-left d-flex">
      <div class="d-flex align-items-center my-auto">
        <h4 class="card-title px-1 fw-bold">{{ subject.name }}</h4>
        <i class="bi bi-check2-square text-success mx-1"
          v-if="isCertified(subject.id)"></i>
      </div>
    </div>
  </div>
</div>
```

Принципот за повлекување на податоците е исто формулиран како и кај

Картичките од професии.

Компонентата <SubjectCheck> од кои е изградена листата на предмети во приказот за додавање на нова професија е имплементирано на следниот начин, прикажано во следниот код:

```
<div v-for="subject in subjects" :key="subject.id" class="col-md-4">
  <div class="d-flex py-3 px-2 rounded-4 border border-2">
    <div class="mx-4 m-auto justify-content-center align-items-center">
      <div>
        <i class="bi bi-check-square text-primary"
          v-if="selectedSubjects.includes(subject.id)"
          @click="toggleSubjectSelection(subject)"></i>
        <i class="bi bi-slash-square text-gray"
          v-else
          @click="toggleSubjectSelection(subject)"></i>
      </div>
    </div>
  </div>
  ...
</div>
```

```

...
<div class="card-body body-line-left d-flex">
  <div class="d-flex align-items-center my-auto">
    <h4 class="card-title px-1"><b>{{ subject.name }}</b></h4>
  </div>
</div>
</div>
</div>

```

Методот со кој се избираат задолжителни предмети за новата професија е

прикажан во следниот блок код:

```

toggleSubjectSelection(subject) {
  const subjectIndex = this.selectedSubjects.indexOf(subject.id);
  if (subjectIndex === -1) {
    this.selectedSubjects.push(subject.id);
    this.selectedSubjectNames.push(subject.name);
  } else {
    this.selectedSubjects.splice(subjectIndex, 1);
    const nameIndex = this.selectedSubjectNames.indexOf(subject.name);
    if (nameIndex !== -1) {
      this.selectedSubjectNames.splice(nameIndex, 1);
    }
  }

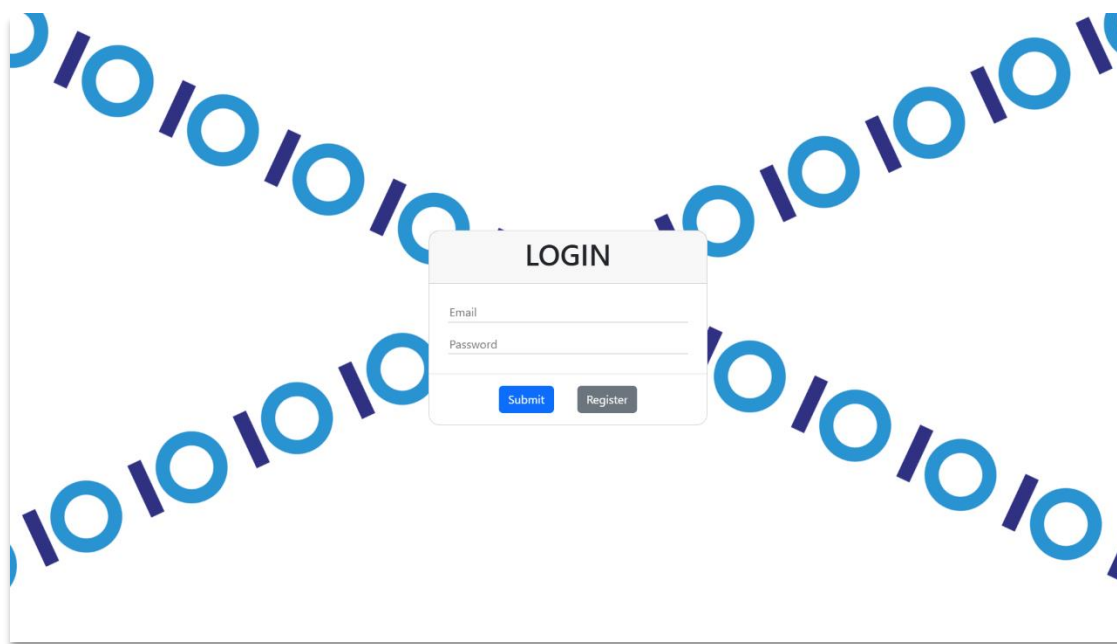
  this.$emit('update-selected-subjects', {
    selectedSubjects: this.selectedSubjects,
    selectedSubjectNames: this.selectedSubjectNames,
  });
},

```

Најава и Регистрација

Веб-апликацијата ФИНКИ Форум и Студиски Водич поддржува најава на корисници користејќи го системот за автентикација на Firebase.

Приказот на формата за најава на корисници е прикажан на Слика 10:

The image shows a login form titled "LOGIN" centered on a white background. The form has two input fields: "Email" and "Password". Below the "Password" field are two buttons: a blue "Submit" button and a grey "Register" button. The background of the entire image is decorated with a pattern of blue and dark blue circles and lines, resembling a stylized DNA helix or a network diagram.

Слика 10. Приказ на форма за најава на корисници.

Firebase Authentication води сметка за корисниците кој се најавуваат на системот, има сопствена база за проектот и врши верификација на влезот на податови.

Самиот систем има неколку барања и закони за автентикација, а тие се:

- Емаил полето треба да биде пополнето со валидна вредност, но не и верифицирана, што значи дека податоците во ова поле задолжително треба да имаат форма на целосна емаил адреса, составена од назив, симболот „@“, назив на провајдер и наставката „.com“.
- Лозинката има основно барање да биде со должина поголема од 5 карактери. Не се поставени останатите основни барања како: потреба од голема буква, потреба од специјален знак и потреба од нумеричка вредност.

Имплементација

Имплементацијата на формата за најава е прикажана во следниот блок код:

```
<div class="card rounded-4 m-auto" style="width: 24rem;">
  <div class="card-header">
    <h1 class="card-title text-center user-select-none">LOGIN</h1>
  </div>
  <ul class="list-group list-group-flush">
    <li class="list-group-item">
      <input class="border-bottom border-2 my-2"
        type="text" placeholder="Email" v-model="email">
      <input class="border-bottom border-2 my-2"
        type="password" placeholder="Password" v-model="password">
    </li>
    <li class="list-group-item mx-auto text-danger text-center my-0"
      v-if="errorMsg">
      <i class="bi bi-exclamation-diamond"></i>
      <p>{{ errorMsg }}</p>
    </li>
  </ul>
  <div class="card-body mx-auto">
    <button class="btn btn-primary mx-3"
      @click="register">Submit</button>
    <button class="btn btn-secondary mx-3"
      @click="redirectRegister">Register</button>
  </div>
</div>
```

Системот го користи вградениот метод за најава на корисникот на Firebase, а тоа е методот `signInWithEmailAndPassword()`.

```
const register = () => {
  signInWithEmailAndPassword(getAuth(), email.value, password.value)
    .then((data) => {
      console.log("Successfully signed in!");
      router.push('/home');
    })
    .catch((error) => {
      console.log(error.code);
      switch (error.code) {
        case "auth/invalid-email":
          errorMsg.value = "Invalid email";
          break;
        case "auth/user-not-found":
          errorMsg.value = "No existing account with this email";
          break;
        case "auth/wrong-password":
          errorMsg.value = "Incorrect password";
          break;
        default:
          errorMsg.value = "Invalid email or password";
          break;
      }
    });
};
```