

Гимназија „Светозар Марковић“  
Ниш

## **МАТУРСКИ РАД**

из предмета Рачунарство и информатика

# **СИСТЕМ ЗА УПРАВЉАЊЕ ПОРУЏБИНАМА У ШКОЛИ**

Ментор:  
Зорана Јанчић

Ученик:  
Матеја Станковић, IV-8

Ниш, мај 2025.





## Садржај

1.	Увод .....	1
2.	Опис проблема и циљеви апликације .....	3
3.	Коришћене технологије.....	5
4.	Структура базе .....	13
5.	Функционалности апликације .....	15
6.	Могућност даљег проширења .....	19
7.	Закључак .....	21
8.	Литература .....	23



## 1. Увод

У савременом образовном окружењу, потреба за ефикасном организацијом и праћењем поруџбина у студентским киосцима или кантинама постаје све израженија. Радници често немају довољно времена за ручну евиденцију, што може довести до грешака, несташица производа или дужих чекања. Овај рад приказује развој web апликације која омогућава једноставно, дигитално управљање поруџбинама студената.

У време дигитализације, ефикасно вођење евиденције и организација пословања представљају кључне аспекте у свакодневном функционисању школских установа. Школски киоск, иако често једноставан по структури, представља важан део инфраструктуре који омогућава ученицима лак приступ основним потроштинама током дана. Због тога је осмишљена апликација која дигитализује процесе уноса, обраде и евиденције поруџбина и производа у оквиру школског киоска.



## 2. Опис проблема и циљеви апликације

Традиционално вођење киоска подразумева ручну евиденцију поруџбина и производа, што може довести до грешака, губитка података, кашњења и непотребних административних проблема. Сврха ове апликације је да омогући једноставно, прегледно и ефикасно управљање киоском, уз јасно раздвојене улоге корисника (админ и радник), аутоматизовано смањење залиха и креирање извештаја.

Циљ овог рада је развијање софтверског решења које ће омогућити корисницима да лако управљају поруџбинама, а администраторима да прате лагер производа, управљају корисницима и анализирају најтраженије артикле путем статистике.





### 3. Коришћене технологије

У изради ове апликације смо користили разне технологије.

Пре свега, за програмски језик у којем је писана апликација, коришћен је Python (Flask framework) за back-end логику. У фајлу app.py се налазе руте и оно што би требало свака од тих функција да ради приликом позива. Ево једног примера:

```
@app.route('/dodaj_korisnika', methods=['GET', 'POST'])
def dodaj_korisnika():
    if session.get('uloga') != 'admin':
        return redirect(url_for('index'))
    if request.method == 'POST':
        ime = request.form['ime']
        email = request.form['email']
        lozinka = generate_password_hash(request.form['lozinka'])
        uloga = request.form['uloga']
        conn = get_db()
        c = conn.cursor()
        c.execute("INSERT INTO korisnici (ime, email, lozinka,
uloga) VALUES (?, ?, ?, ?)", (ime, email, lozinka, uloga))
        conn.commit()
        conn.close()
        return redirect(url_for('korisnici'))
    return render_template('dodaj_korisnika.html')
```

У првој линији је дефинисана рута додај корисника. И на ту руту се може доћи и get и post захтевом. Уколико није у питању корисник који је логован и пријављен као admin онда се претпоставља да он и нема привилегије да додаје кориснике. Онда се тражи да метод буде post и из request-а узимамо име, емаил, лозинку и улогу и инсертујемо у базу. Након чега комитујемо наш упит и затварамо конекцију ка бази, и редиректујемо страницу на приказ корисника.

Као базу података смо користили SQLite. Сама база је смештена у једном фајлу и тај фајл се зове baza.db. Коришћење базе ћемо објаснити у једном од наредних поглавља. У суштини база има следеће табеле:

- Korisnici – овде смештамо кориснике нашег система. Свако од корисника има своје име, пријављује се маилом и пасвордом и има своју улогу. Две су могуће улоге радник и админ. Админ може

да мења кориснике и додаје производе, док радник може да прави само поруџбине.

- `Proizvodi` – где смештамо производе, заправо њихове називе, цену и количину.
- `Porudzbine` – у ову табелу смештамо опште податке о поруџбини, а то су датум поруџбине, име ученика који је направио поруџбину и ид корисника који формира поруџбину из програма.
- `Stavke_porudzbine` – где имамо прво поље ид поруџбине које означава на коју поруџбину се односи тај ред, производ ид које означава који производ је у поруџбини, количина и цена по јединици мере.

Следећа ствар коју смо користили јесте HTML, CSS и Bootstrap за преглед страница. На свакој страници је укључен Bootstrap и JQuery као библиотеке за изглед и функционисање елемената на страници, као што су дугмићи, обавештења, хедери и остало. Ево једног примера:

```
<!DOCTYPE html>
<html>
<head>
  <title>Dodaj proizvod</title>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.m
in.css" rel="stylesheet">
</head>
<body class="container mt-4">
  <h2>Dodaj proizvod</h2>
  <form method="post" class="w-50">
    <div class="mb-3"><label>Naziv:</label><input type="text"
name="naziv" class="form-control" required></div>
    <div class="mb-3"><label>Cena:</label><input type="number"
name="cena" step="0.01" class="form-control" required></div>
    <div class="mb-3"><label>Količina:</label><input type="number"
name="kolicina" class="form-control" required></div>
    <button type="submit" class="btn btn-primary">Dodaj</button>
    <a href="{{ url_for('proizvodi') }}" class="btn btn-
secondary">Nazad</a>
  </form>
</body>
</html>
```

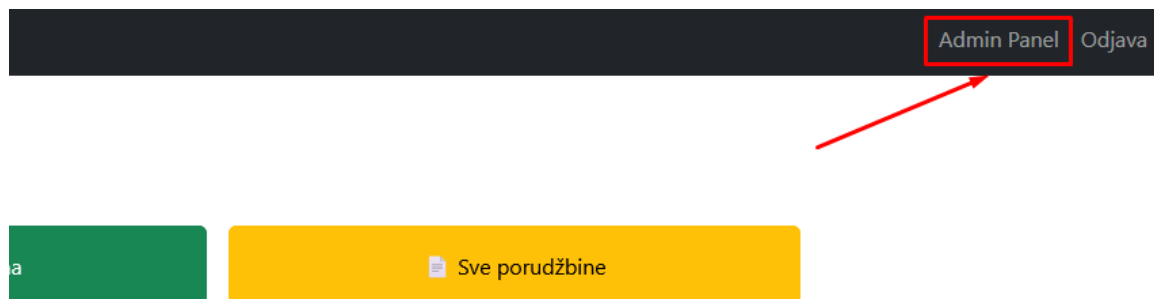
Овде имамо на почетку укључен стандардни хедер. Линија која укључује Bootstrap је видљива и ту се укључује Bootstrap директно са сајта. Могуће је и скинути локално, па га укључити.

Контејнери су најосновнији елемент распореда у Bootstrap-у и неопходни су када се користи наш подразумевани систем мреже. Контејнери се користе за смештање, попуњавање и (понекад) центрирање садржаја унутар њих. Иако контејнери могу бити угнежђени, већина распореда не захтева угнежђени контејнер.

Након контејнера иде таг за форму која има метод пост, што значи да ће се форма прослеђивати тим методом и линк у делу где стоји адреса неће садржати додатне променљиве.

Након тога иду улазна поља типа назив, цена, количина која служе за унос тих ствари о производу и на крају два линка Додај и Назад. Кликом на Додај уносимо нови производ.

Део који контролише понашање одређених делова на страници је JQuery. За тај део ћемо објаснити функционисање руте `admin_dashboard`. Кренимо од саме руте у фајлу `app.py`. Линк који нас води на ову руту се појављује админ кориснику, када је улогован у горњем десном углу програма.



Када кликнемо на тај линк одлазимо на следећу страницу:

## Administratorski Panel

Ukupno porudžbina  
3

Ukupna vrednost  
23600.0 RSD

Top proizvod  
Sundjer

Korisnici

Proizvodi

Dodaj korisnika

Ime	Email	Uloga	Akcije
Mateja	mateja@mail.com	admin	Obrisi
Sima	sima@mail.com	radnik	Obrisi

Где у гоњем делу имамо колико је укупно поруџбина било до сада, колика је укупна вредност тих поруџбина и који је производ најпопуларнији. Док у доњем делу имамо два таба Корсиници и Производи где добијамо преглед корисника и преглед производа. У могућности смо да додамо новог корисника или обришемо постојећег.

## Administratorski Panel

Ukupno porudžbina  
3

Ukupna vrednost  
23600.0 RSD

Top proizvod  
Sundjer

Korisnici

Proizvodi

Dodaj proizvod

Naziv	Cena	Količina	Akcije
Flomaster	50.0	994	Izmeni Obrisi
Bela tabla 120x120	5000.0	6	Izmeni Obrisi
Sundjer	200.0	1988	Izmeni Obrisi
Kreda	100.0	991	Izmeni Obrisi

На другој картици Производи нам је списак свих производа са ценама, количинама и акцијама за измену или брисање производа. Рута која нас води до овог менија админ је:

```
@app.route('/admin')
def admin():
    if session.get('uloga') != 'admin':
        return redirect(url_for('index'))
    conn = get_db()
    c = conn.cursor()
    c.execute("SELECT * FROM korisnici")
    korisnici = c.fetchall()
```

```

c.execute("SELECT * FROM proizvodi")
proizvodi = c.fetchall()
c.execute("SELECT COUNT(*) FROM porudzbine")
ukupno_porudzbina = c.fetchone()[0]
c.execute("SELECT SUM(s.kolicina * s.cena_po_jedinici) FROM
stavke_porudzbine s")
ukupna_vrednost = c.fetchone()[0] or 0
c.execute("SELECT p.naziv FROM proizvodi p JOIN stavke_porudzbine s
ON s.proizvod_id = p.id GROUP BY p.id ORDER BY SUM(s.kolicina) DESC
LIMIT 1")
top_proizvod = c.fetchone()
statistika = {
    'ukupno_porudzbina': ukupno_porudzbina,
    'ukupna_vrednost': round(ukupna_vrednost, 2),
    'top_proizvod': top_proizvod[0] if top_proizvod else 'N/A'
}
conn.close()
return render_template("admin_dashboard.html", korisnici=korisnici,
proizvodi=proizvodi, statistika=statistika)

```

Прво се проверава да ли је пријављени корисник админ или није. Ако није онда му се не дозвољава приступ. Након тога се повезујемо са базом и отварамо курсор и извршавамо упите који нам довлаче све кориснике, производе, поруџбине и најтраженије производе. На крају се позива рендер странице са променљивама чије смо вредности добили.

```

<!DOCTYPE html>
<html>
<head>
    <title>Admin Panel</title>
    <meta charset="UTF-8">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.m
in.css" rel="stylesheet">
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bun
dle.min.js"></script>
</head>
<body>
<div class="container mt-4">
    <h2 class="mb-4">Administratorski Panel</h2>

    <!-- Statistika -->
    <div class="row mb-4">
        <div class="col-md-4">
            <div class="card text-bg-primary mb-3">
                <div class="card-body">
                    <h5 class="card-title">Ukupno porudžbina</h5>
                    <p class="card-text fs-4">{{
statistika.ukupno_porudzbina }}</p>
                </div>
            </div>

```

```

</div>
<div class="col-md-4">
  <div class="card text-bg-success mb-3">
    <div class="card-body">
      <h5 class="card-title">Ukupna vrednost</h5>
      <p class="card-text fs-4">{{
statistika.ukupna_vrednost }} RSD</p>
    </div>
  </div>
</div>
<div class="col-md-4">
  <div class="card text-bg-warning mb-3">
    <div class="card-body">
      <h5 class="card-title">Top proizvod</h5>
      <p class="card-text fs-4">{{
statistika.top_proizvod }}</p>
    </div>
  </div>
</div>
</div>
<!-- Tabovi -->
<ul class="nav nav-tabs" id="adminTab" role="tablist">
  <li class="nav-item" role="presentation">
    <button class="nav-link active" id="korisnici-tab" data-bs-
toggle="tab" data-bs-target="#korisnici" type="button"
role="tab">Korisnici</button>
  </li>
  <li class="nav-item" role="presentation">
    <button class="nav-link" id="proizvodi-tab" data-bs-
toggle="tab" data-bs-target="#proizvodi" type="button"
role="tab">Proizvodi</button>
  </li>
</ul>
<div class="tab-content mt-3">
  <div class="tab-pane fade show active" id="korisnici"
role="tabpanel">
    <a href="{{ url_for('dodaj_korisnika') }}" class="btn btn-
success btn-sm mb-3">Dodaj korisnika</a>
    <table class="table table-striped">

<thead><tr><th>Ime</th><th>Email</th><th>Uloga</th><th>Akcije</th></tr>
</thead>

    <tbody>
      {% for k in korisnici %}
      <tr>
        <td>{{ k[1] }}</td><td>{{ k[2] }}</td><td>{{ k[4]
}}</td>
        <td>
          <a href="{{ url_for('obrisi_korisnika',
id=k[0]) }}" class="btn btn-danger btn-sm" onclick="return
confirm('Obrisati korisnika?') ">Obriši</a>
        </td>
      </tr>
      {% endfor %}
    </tbody>
  </table>

```

```

</div>
<div class="tab-pane fade" id="proizvodi" role="tabpanel">
  <a href="{{ url_for('dodaj_proizvod') }}" class="btn btn-
success btn-sm mb-3">Dodaj proizvod</a>
  <table class="table table-striped">

<thead><tr><th>Naziv</th><th>Cena</th><th>Količina</th><th>Akcije</th><
/tr></thead>

  <tbody>
    {% for p in proizvodi %}
    <tr>
      <td>{{ p[1] }}</td><td>{{ p[2] }}</td><td>{{ p[3]
}}</td>

      <td>
        <a href="{{ url_for('izmeni_proizvod',
proizvod_id=p[0]) }}" class="btn btn-warning btn-sm">Izmeni</a>
        <a href="{{ url_for('obrisi_proizvod',
proizvod_id=p[0]) }}" class="btn btn-danger btn-sm" onclick="return
confirm('Obrisati proizvod?') ">Obriši</a>
      </td>
    </tr>
    {% endfor %}
  </tbody>
</table>
</div>
</div>
</body>
</html>

```

Овако изгледа HTML за страницу админ. Прво ће нам се приказати админ панел са информацијама о Укупно поруџбина, Укупној вредности и Топ производу. А након тога табови са Корисницима и Производима. Оно што направи табове јесте JQuery од ове ul листе. Оно што би овде могли још да нагласимо јесте for петља у HTML-у коју видимо на пример у

```
{% for k in korisnici %}
```

где се у променљивој k стављају чланови низа корисници.

На страници где можемо прегледати све поруџбине имамо и Python CSV за извоз података у функцијама `izvestaj_csv` и `izvoz_porudzbina`. У њима се након упита за прибављање података из базе формира прост excel фајл са насловним редом и табелом испод.





## 4. Структура базе

База података садржи табеле: корисници, производи, поруџбине и ставке поруџбине. Свака поруџбина садржи више ставки које повезују одређени производ и количину. Оно што је јединствено за сваку поруџбину јесте датум, ученик који је захтевао поруџбину, као и ид корисника који уноси саму поруџбину. Од ставки поруџбине имамо ид, ид поруџбине, ид производа који је одабран у поруџбини, количину и цену.

Систем омогућава вођење корисничких налога, где се свака акција бележи са корисником који је извршио поруџбину. У табели корисници имамо поља ид, име, које представља име и презиме самог корисника, емаил адресу коју корисник користи за пријављивање, и она мора бити јединствена, лозинку, која се хешира приликом убацивања у базу и чува се таква хеширана лозинка. Не чувамо лозинку која је унета у базу. У табели корисници се на крају налази и поље улога које може бити админ или радник. Админ има додатне функционалности да види админ панел који смо описали у претходном поглављу.

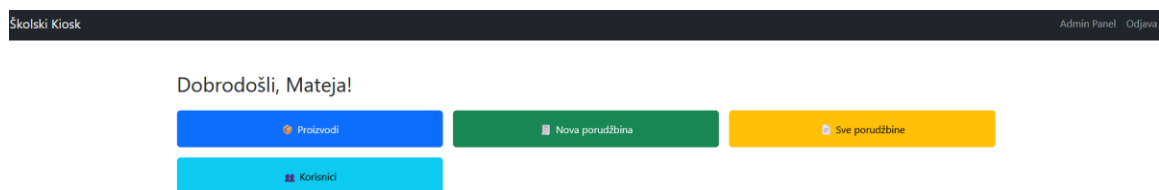
Табела производи има поља ид, назив који мора бити јединствен, цену и количину. Овде се води регистар о томе колико ког производа имамо још увек на стању.

Постоје још две табеле у нашој бази које служе за иницијално креирање наших табела, заједно са упитима који се том приликом употребљавају. То су табеле `sqlite_master` и `sqlite_sequence`.



## 5. Функционалности апликације

Од самих функционалности апликације пре свега долазимо на страницу за пријављивање која има два поља емаил и лозинку. Корисник се исправним уносом пријављује на апликацију. Уколико лозинка није исправна, ситем га неће пријавити на апликацију. Од улога корисника имамо админ и радник. Админ, након пријављивања у горњем десном углу има линк ка Админ делу који смо описали у претходним поглављима. Што се тиче осталих радњи имамо следећи екран:



Овде видимо да корисник има могућност да оде и дода производ, да на тој страници има још и линкове ка измени назива производа и за брисање.

### Lista proizvoda

[Dodaj novi proizvod](#)

Naziv	Cena	Količina	Akcije	
Flomaster	50.0	990	<a href="#">Izmeni</a>	<a href="#">Obriši</a>
Bela tabla 120x120	5000.0	0	<a href="#">Izmeni</a>	<a href="#">Obriši</a>
Sundjer	200.0	1917	<a href="#">Izmeni</a>	<a href="#">Obriši</a>
Kreda	100.0	954	<a href="#">Izmeni</a>	<a href="#">Obriši</a>

[Početna](#)

Има могућност уноса нове поруџбине:









## Nova porudžbina

Učenik:

Datum:

06 / 01 / 2025



Proizvod	Cena	Dostupno	Količina
Flomaster	50.0	990	0  
Bela tabla 120x120	5000.0	0	0  
Sundjer	200.0	1917	0  
Kreda	100.0	954	0  

Završi porudžbinu

Početna

Нову поруџбину уноси тако што унесе име ђака који је поруџбину захтевао, датум поруџбине, као и од доступних производа бира количину коју ће да унесе. Завршетак уноса поруџбине се задаје на дугме Заврши поруџбину.

Имамо још и дугме за преглед свих поруџбина. На тој страници имамо филтер од датума до датума и ученика који је захтевао поруџбину, па можемо тако исфилтрирати.

## Porudžbine

Od datuma:

mm / dd / yyyy



Do datuma:

mm / dd / yyyy



Učenik:

Filtriraj

Poništi

Preuzmi sve porudžbine (CSV)

Najtraženiji proizvodi (CSV)

Datum	Učenik	Radnik	Akcija
2025-06-01	Sima Simić	Mateja	<a href="#">Obriši</a>
2024-01-01	Pera	Mateja	<a href="#">Obriši</a>
2024-01-01	Mika	Mateja	<a href="#">Obriši</a>
2024-01-01	Jelena	Mateja	<a href="#">Obriši</a>
2024-01-01	Petar Peric	Mateja	<a href="#">Obriši</a>

Početna

Оно што је важно на овој страници јесте и дугме Преузми све поруџбине (цсв) где у MS Excel-у добијамо списак свих поруџбина које су направљене, а на дугме Најтраженији производ добијамо списак производа са највише наруџбина.

И на крају дугме Корисници, чију смо функцију раније описали.

Можемо закључити да је кориснички систем апликације веома једноставан и прегледан, са главним опцијама приказаним путем навигације. Након пријаве, корисници имају приступ функцијама према својој улози.

Приступ апликацији је заштићен логин формом са шифрованим лозинкама помоћу 'werkzeug.security' модула. Улоге корисника осигуравају да само админ може приступити администрацији и изменама података о корисницима.



## 6. Могућност даљег проширења

У будућности апликација се може проширити увођењем QR кодова, уплатницама за плаћање наруџбина, повезивањем са мобилним апликацијама, као и праћењем потрошње по одељењима. Такође, може се увести штампа рачуна или напреднији систем претраге и филтрирања. Од мање крупних ствари требало би да обезбеди да корисник не може да обрише самог себе, да се не може додати производ два пута са истим именом, као и на страници за креирање понуде где видимо све производе, па би у школском киоску где је могуће наручити више производа било незгодно правити наруџбину са великим бројем производа.

Додавање лог активности, у смислу који корисник је шта урадио, и бележење грешака које прави би био још један аспект у даљем развоју апликације.





## 7. Закључак

Развој ове апликације представља значајан корак ка дигитализацији процеса у образовним установама. Уз једноставан интерфејс, сигурносне мере и могућности извештавања, систем омогућава ефикасан и прегледан рад у свакодневним активностима школског киоска. Његова флексибилност и могућности даљег развоја чине га корисним алатом за све сличне намене.

Развијена апликација је функционална, једноставна за коришћење и омогућава ефикасно управљање школским киоском. Поред функционалне користи, апликација пружа и едукативну вредност кроз употребу савремених технологија у програмирању.



## 8. Литература

- [1] Сајт [flask.palletsprojects.com](https://flask.palletsprojects.com/), о Flask-у, увођење, инсталација и коришћење
- [2] Сајт [www.sqlite.org](https://www.sqlite.org). Све о SQLite-у
- [3] Сајт [getbootstrap.com](https://getbootstrap.com), све о bootstrap-у
- [4] Сајт [jquery.com](https://jquery.com), све о jQuery-ју.
- [5] Eric Matthes, *Python Crash Course*, No Starch Press, 2022.



Датум предаје:

\_\_\_\_\_

Чланови испитне комисије:

председник \_\_\_\_\_

испитивач \_\_\_\_\_

члан комисије \_\_\_\_\_

КОМЕНТАР:

Оцена вредности матурског рада \_\_\_\_\_

Оцена одбране матурског рада \_\_\_\_\_

Закључна оцена \_\_\_\_\_

Датум одбране:

\_\_\_\_\_