



## ВЕБ АПЛИКАЦИЈА ЗА ПРОДАЈУ ОНЛАЈН КУРСЕВА WEB SHOP FOR ONLINE COURSES

Тамара Јовичић, Факултет техничких наука, Нови Сад

### Област – РАЧУНАРСТВО И АУТОМАТИКА

**Кратак садржај** – У раду је описан систем за куповину онлајн курсева. Детаљно је описан модел поменутог система кроз дијаграме. Описана је имплементација система и представљен је изглед апликације. Анализирана су слична решења.

**Кључне речи:** е-плаћање, е-трговина, курсеви, PayPal, AWS

**Abstract** – The paper describes the system for purchasing online courses. It is described in detail through diagrams. The implementation of the system is described and the appearance of the application is presented. Similar solutions were analyzed.

**Keywords:** e-commerce, e-shopping, courses, PayPal, AWS

### 1. УВОД

Е-трговина подразумева процесе продаје и куповине добара и услуга коришћењем савремених комуникационих технологија и Интернета. Модерне платформе за е-трговину су софистициране апликације које обављају вишеструке функције [1].

Системи електронског плаћања могу помоћи корисницима да уштеде своје време и новац. У односу на традиционалне моделе плаћања, у системима електронског плаћања потребно је веће поверење и прихватање чињенице да купац не може уживо да види производ у тренутку куповине. Са електронским плаћањем, трансакције се могу обавити брзо, што представља једну од суштинских предности плаћања.

### 2. ПРЕГЛЕД ПОСТОЈЕЋИХ РЕШЕЊА

Плаћање и куповина посредством Интернета је данас веома заступљено. Е-учење је доживело велики раст последњих година, поготово од 2020. године када је почела пандемија COVID-19 [3]. Постоји велики број веб апликација које се баве продајом различитих ресурса, такође, постоји и велики број веб продавница које се баве онлајн продајом курсева [4]. У наставку ће бити описани неки од система који су направљени у те сврхе и који су данас опште познати.

### НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Горан Сладић, ред. проф.

### 2.1. Udemy

Један од најпознатијих платформи за куповину и преглед курсева јесте управо Udemy. Udemy има далеко највише доступних курсева у поређењу са другим платформама за учење на мрежи, нудећи 183.000 курсева [4]. Садржи преко 65.000 предавача, курсеви су доступни на око 65 језика [4]. Ова образовна платформа има и преко 40 милиона студената, при чему се тај број свакодневно повећава. Python је главни програмски језик који се користи на страни сервера. Међутим, Удему је почео са PHP-ом као главним програмским језиком на бекенд страни [5]. На клијентској страни, AngularJS игра главну улогу. За претрагу и аналитику користи се Elasticsearch. Да би се побољшале свеукупне перформансе веб странице, коришћени су Memcached, Zencoder, и Redis, због чега се у Udemy-у не примењују кашњења и успореност [5].

### 2.2. Coursera

Coursera је онлајн образовна платформа са 23 милиона корисника, која је посвећена понуди висококвалитетних онлајн курсева обуке широм света [6]. Док је у партнерству са универзитетима и предузећима светске класе, пружа студентима прилику да добију сертификате од реномираних институција након што одслушају њихове курсеве.

Апликација претежно написана у Python-у и Scala-и са фрејмворцима Play, Django и Scalding. Као и Udemy, Coursera претежно користи MySQL за управљање релационом базом података. На клијентској страни, претежно се користи JavaScript, тачније ReactJS. Такође, користи се и Bootstrap, html и css [7].

### 2.3. Skillshare

SkillShare је још једна позната платформа која има више од 4 милиона студената и до 24.000 курсева у различитим дисциплинама [8]. Часови курса трају од 20 до 60 минута и направљени су од низа малих видео снимака, школских пројеката и студентске заједнице која подстиче интеракцију кроз различите форуме за дискусију.

На серверској страни, највише се користи PHP [9]. Клијентска страна кода написана је користећи JavaScript са TypeScript-ом. Користе се библиотеке React, jQuery, D3.js, Backbone.js и многе друге. Што се тиче базе података, као и претходно две наведене платформе, користи се MySQL, користе се и неки од AWS сервиса, као што су Amazon S3 и Amazon Redshift [9].

### 3. СПЕЦИФИКАЦИЈА СИСТЕМА

У односу на анализе платформи које су описане у претходном поглављу, формирана је спецификација, а затим и архитектура система за куповину онлајн курсева.

Идеја овог система јесте да корисницима обезбеди преглед курсева, куповину и преузимање истих. Постоји могућност куповине једног курса, али и серије курсева.

Омогућено је и плаћање посредством *PayPal*-а, затим могућност претплате на месечни период у одређеном износу. Један од задатака је и заштита садржаја, отежавање преузимања материјала те је убачен *watermarking* на свим материјалима. Апликација се развија уз основац на *AWS* цлоуд платформу, те је коришћен *AWS S3* сервис за складиштење видео материјала, слика и других материјала које курсеви садрже.

Систем за онлајн куповину курсева намењен је за три врсте корисника: купца, продавца и администратора.

#### 3.1. Активности у систему

Сва три корисника, купац, продавац и администратор, имају могућност пријављивања у систем. Приликом пријављивања, уноси се и-мејл адреса и лозинка. Постоји могућност и регистрације корисника у систем. Када се корисник региструје, уносе се његови подаци и тај корисник аутоматски добија улогу купца. Продавци се додају у систем од стране администратора, а администратори се директно додају у базу података.

Корисници могу да мењају своје личне податке. Такође, могу и да ресетују своју лозинку, уколико су је заборавили или слично. Администратор има могућност да забрани неком од корисника да се опет пријављује у систем уколико то жели.

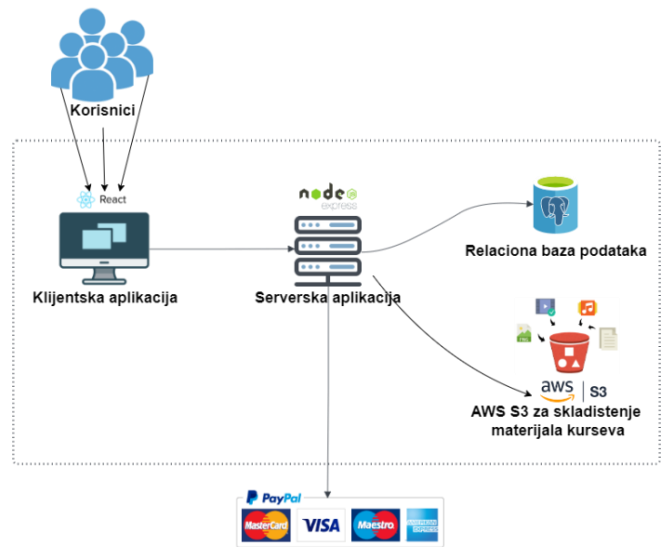
Једна од главних активности у систему је да корисници могу да прелиставају постојеће курсеве на почетној страници. Купац може да додаје курсеве у своју корпу, а исто тако и да уклања курсеве из ње. Уколико је сигуран да жели одређени курс да купи, онда може да изврши плаћање. Када је неки од курсева плаћен, купац добија могућност да види сав садржај тог курса и да преузме материјале на свој уређај.

Курсеве у систем додаје корисник који има улогу продавца. Приликом додавања, уноси назив курса, опис, цену и додаје материјале тачније пдф-ове, слике, видео записе и друго. Продавац у било ком тренутку може да измени податке о курсевима које је додао. Администратор исто има могућност да управља курсевима који се налазе у систему.

#### 3.2. Архитектура система

На слици 1. визуелно је представљена архитектура система. Главне делове система чини једна клијентска апликација и једна серверска, које међусобно комуницирају и размењују податке.

Урађена је интеграција са *PayPal*-ом путем кога се остварује опција плаћања.



Слика 1. Архитектура система

За складиштење података о кориснику, трансакцијама и курсевима користи се *PostgreSQL* база података. Додатно, апликација се ослања и на *AWS S3* сервис, где се складиште сви додатни материјали везани за курсеве, а који нису претходно сачувани у релационој бази.

Клијентска апликација представља графички интерфејс и помоћу ње, корисници виде шта систем нуди од функционалности. Написана је у *JavaScript*-у, при чему је коришћена библиотека *React*.

Серверску страну чини монолитна апликација у којој је имплементирана сва пословна логика потребна за реализацију једног система за плаћање. Код бекенд стране се такође ослања на програмски језик *NodeJS*. Што се тиче базе података, коришћен је *PostgreSQL* објектно-релациони систем за управљање базама података. Осетљиви подаци, као на пример што су шифре корисника, су заштићени користећи *bcrypt* механизма за хеширање. Овај систем се ослања такође и на цлоуд платформу *Amazon Web Services (AWS)*, тачније на *S3* сервис, који омогућава чување различитог садржаја, као на пример видео материјала, пдф докумената, слика итд.

#### 3.3. Модел података

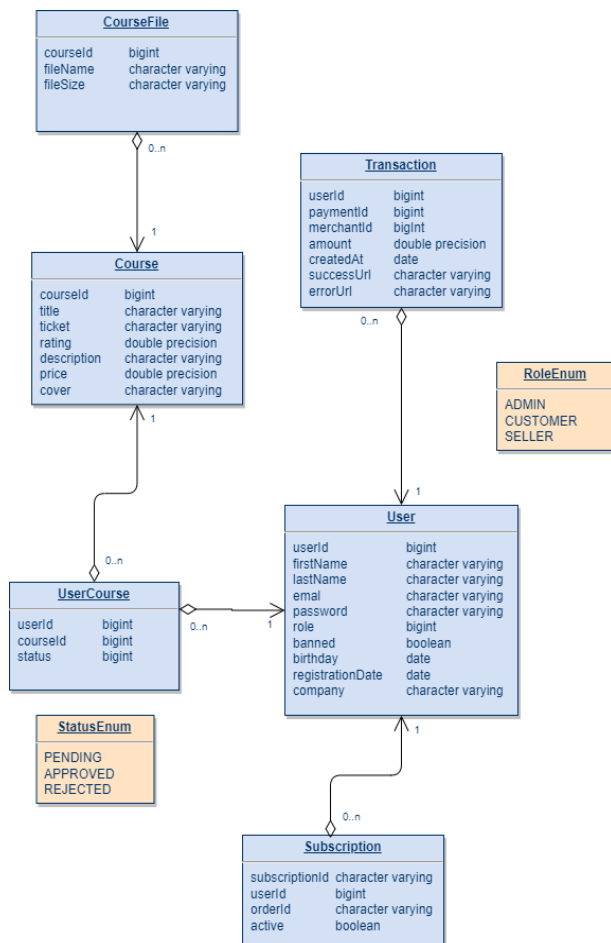
Подаци у систему су структурирани, тако да је релациона база података погодна за њихово складиштење. На слици 2. приказан је дијаграм класа које се користе у систему за продају онлајн курсева. Најбитније класе система су:

- корисници,
- курсеве и
- трансакције.

### 4. ИМПЛЕМЕНТАЦИЈА

#### 4.1. Интеграција са *PayPal*-ом

*NodeJS* се ослања на *third party* библиотеку која се користи за интеграцију са *PayPal*-ом. Та библиотека је *paypal-rest-sdk*.



Слика 2. Класни дијаграм

Пре него што се апликација повеже са *PayPal*-ом, потребно је направити *PayPal developer* налог на следећем линку: <https://developer.paypal.com/home>. Затим је потребно направити нову апликацију како би се добили креденцијали потребни за повезивање система са *PayPal*-ом, тачније *client\_id* и *client\_secret*. Поред претходног налога који је направљен, потребно је направити и *sandbox* налог, који ће се користити приликом тестирања плаћања.

Када корисник кликне на дугме за плаћање курсева, захтев се шаље на ендпоинт `"/pay"`. *Body* овог захтева садржи све ставке које корисник жели да плати и износ трансакције.

Да би се трансакција извршила, потребно је креирати *JSON* објекат *create\_payment\_json* који садржи информације о сврси уплате, платиши, дати су *URL*-ови за редирекцију, садржај трансакције, односно све ставке које се плаћају, и износ уплате.

Када корисник одабере начин плаћања и провери све приказане детаље, може да притисне дугме *continue* како би се плаћање реализовало.

Постоје два сценарија која могу да се десе приликом завршетка трансакције. Један је да се трансакција успешно реализује, а други је да дође до неке грешке или да корисник откаже извршење исте.

Кад год је трансакција успешна, требало би да добијемо детаље трансакције од *PayPal*-а преко методе *paypal.payment.execute()*. Детаљи трансакције *PayPal*-а могу се сачувати у бази података за будуће

референце. Ако је трансакција била успешна, кориснику ће бити приказана страница о успеху. Уколико је трансакција прекинута, ништа од података се не чува у базу података, а корисник бива редиректован до странице за прекинуту трансакцију. У систему је омогућена и **претплата** преко *PayPal*-а која функционише на следећи начин:

1. Креирати производ који представља робу или услугу.
2. Направити план који ће представљати циклусе плаћања за вашу претплату.
3. Користити *PayPal JavaScript SDK* за представљање дугмета. Када купац одабере дугме, почиње искуство претплате.
4. Купац треба да се сложи и претплаћује.
5. Дугме позива *API* за *PayPal* претплате да креира претплату.
6. Купац треба да види потврду претплате.

## 4.2. Интеграција са AWS-ом

Да би се остварила интеграција са *AWS*-ом и да би се направио *S3 Bucket*, потребно је прво направити налог на *Amazon*-у. Процедура регистрације је једноставна, једино што је потребно да се верификује број телефона и унесу подаци о кредитној картици. Након креирања налога, могуће је креирати *S3 Bucket*.

*S3* складишти податке као објекте у *bucket*-у, које креира корисник. *Bucket* је контејнер за објекте ускладиштене у *Amazon S3* сервису. Могуће је складиштење било којег броја објеката. Подаци се могу отпремати, преузимати или брисати путем одговарајућих *RESTful http* захтева.

На *S3 bucket*-у могуће је складиштење фајлова различитих врста формата. За потребе овог система, *S3 bucket* је коришћен за чување садржаја курсева, као што су слике, видео записи, пдф датотеке, аудио звуци и слично.

За креирања *bucket*-а неопходно је унети назив и изабрати регион који је најближи тренутној локацији корисника. Затим је потребно дати дозволу корисницима да приступе овој групи. Тако да треба креирати новог *IAM* корисника и да му дамо дозволу да користи *S3 bucket*. Након успешног креирања *IAM* корисника, *S3 bucket* конфигурација је спремна да би се успоставила интеграција са *NodeJs* апликацијом. Да би се у сервисној *NodeJs* апликацији омогућила интеграција са *AWS*-ом, потребно је инсталирати *npm paket aws-sdk*.

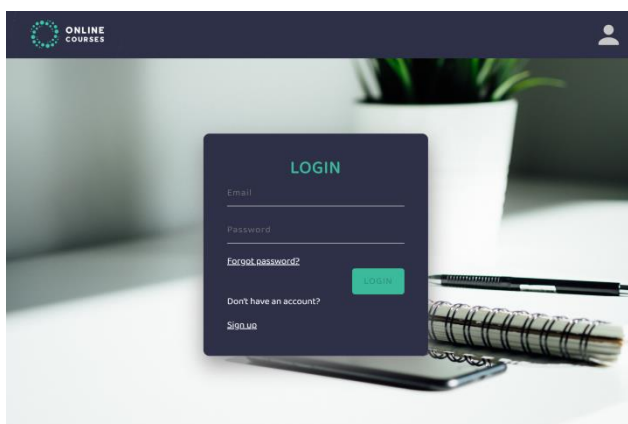
Да би се креирала инстанца *S3* објекта, потребно је користити *AWS SDK*. Варијабле које се користе *AWS\_ACCESS\_KEY\_ID* и *AWS\_SECRET\_ACCESS\_KEY* су добијене приликом креирања *IAM* корисника.

Да би се датотека додала у *S3 bucket* користи се метода *s3.upload()*, а да би се преузеле датотеке користи се *s3.getObject()*.

## 5. ПРИКАЗ ИМПЛЕМЕНТИРАНОГ РЕШЕЊА

Приликом покретања апликације, прва страница која се појави је страница за пријављивање корисника у систем. На слици 3. приказана је форма за логовање

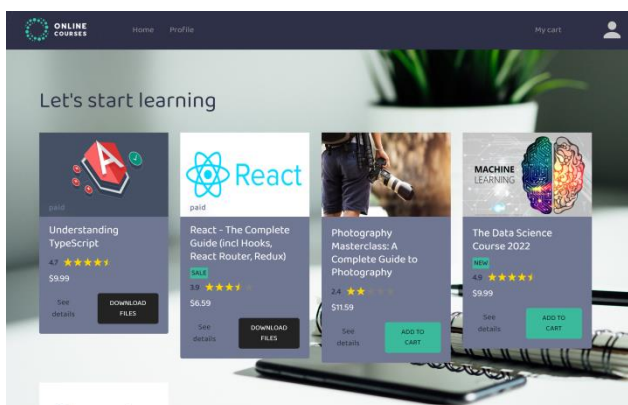
коју корисник треба да попуни како би могао да користи систем.



Слика 3. Пријава корисника у систем

Када се корисник успешно пријави на систем приказује му се почетна страница која је приказана на слици 4. На почетној страници се приказује листа курсева које корисник може да дода у корпу. Уколико је неки од курсева већ купљен, тада је могуће преузимање материјала тог курса. Могуће је видети и додатне детаље курсева уколико се кликне на дугме „See details”.

Са почетне странице корисник може да иде до странице за приказ корисничке корпе и до профила корисника. Дугмићи за прелазак на корисников профил и корпу се налазе у таб менију који се налази на врху странице.



Слика 4. Почетна страница

## 6. ЗАКЉУЧАК

У овом раду детаљно је описано како функционише систем за продају онлајн курсева. Развојем Интернета и информационих технологија, настајала је све већа потреба за електронском трговином и плаћањем. Тако је данас, електронско плаћање опште присутно, те постоји доста апликација која се базирају баш на е-трговину.

У почетном делу овог рада описане су три апликације које спадају у једне од најпопуларнијих апликација које се данас користе за продају онлајн курсева. Засновано по њиховом искуству, настао је и систем

који је тема овог рада. У овом раду описана је спецификација система, изглед целе архитектуре, као и модел система помоћу дијаграма класа и дијаграма случајева коришћења. У делу имплементације, описане су све технологије које су коришћене да се овај систем реализује. Битни делови ове апликације су интеграција са PayPal-ом и AWS S3 bucket-ом, чији је опис такође детаљно дат. Представљени су делови кода који су коришћени да се интеграција изврши. Поред делова кода, дате су и слике које визуелно приказују коришћење PayPal-а и AWS-а. Последње поглавље представља приказ имплементiranог решења, где су дати сцреенсхот-ови, помоћу којих се ближе описује изглед корисничког интерфејса.

## 4. ЛИТЕРАТУРА

- [1] “A system for e-commerce website evaluation”, Snezana Sulova
- [2] “7 Benefits of Electronic Payments”, Nikunj Gundaniya, <https://customerthink.com/7-benefits-of-electronic-payments/>
- [3] “The Transformation of Higher Education After the COVID Disruption: Emerging Challenges in an Online Learning Scenario”. Victor J. García-Morales<sup>1</sup>, Aurora Garrido-Moreno, Rodrigo Martín-Rojas <https://www.frontiersin.org/articles/10.3389/fpsyg.2021.616059/full>
- [4] “Best Online Learning Platforms”, Amanda Prah, <https://www.verywellfamily.com/best-online-learning-platforms-5073725>
- [5] “What programming language Udemy is written in”, Kitty Gupta <https://www.freelancinggig.com/blog/2018/10/23/what-programming-language-udemy-is-written-in/>
- [6] “Crunchbase”, <https://www.crunchbase.com/>
- [7] “Technologies used by Coursera”, <https://techstacks.io/stacks/coursera/>
- [8] “Best Online Learning Platforms (updated 2022)”, Kyriaki Raouna, <https://www.learnworlds.com/online-learning-platforms/>
- [9] “Skillshare tech stack”, <https://himalayas.app/companies/skillshare/tech-stack>

## Кратка биографија:



**Тамара Јовичић** рођена је 14.01.1998. године у Новом Саду. Завршила је основне академске студије 2020. године на Факултету техничких наука. Уписала је мастер студије исте године, студијски програм Електронско пословање.

контакт: taamarajovicic@gmail.com