



## RAZVOJ WEB APLIKACIJE ZA AUTOMATIZACIJU PROCESA POSLOVANJA TURISTIČKE AGENCIJE

### DEVELOPMENT OF WEB APPLICATION FOR TOURIST AGENCY BUSINESS AUTOMATION

Kristina Ševerov, *Fakultet tehničkih nauka, Novi Sad*

#### Oblast – INFORMACIONO-KOMUNIKACIONI SISTEMI

**Kratak sadržaj** – *Ovaj rad predstavlja opis izrade web aplikacije za turističku agenciju, od baze podataka, backend i frontend dela aplikacije kao i samog deployment procesa.*

**Ključne reči:** *Baza podataka, web aplikacija, azure deployment*

**Abstract** – *This paper presents a description of the creation of a web application for a travel agency, from the database, backend and frontend parts of the application, as well as the deployment process itself.*

**Keywords:** *Database, web application, azure deployment*

#### 1. UVOD

Web aplikacija je računarski program koji koristi web preglednike i web tehnologiju za izvršavanje zadataka putem Interneta. Milioni preduzeća koriste Internet kao isplativ komunikacioni kanal. Omogućava im razmenu informacija sa ciljnim tržištem i brže i sigurne transakcije. Međutim, efikasno angažovanje je moguće samo kada je preduzeće u stanju da prikupi i skladišti sve potrebne podatke i ako ima sredstva za obradu ovih informacija i predstavljanje rezultata korisniku [1].

Evo kako izgleda tipičan tok web aplikacija:

- Korisnik šalje zahtev web serveru preko interneta, putem web pregledača ili korisničkog interfejsa aplikacije.
- Web server prosleđuje ovaj zahtev odgovarajućem serveru web aplikacija.
- Server web aplikacija izvršava traženi zahtev - poput upita baze podataka ili obrade podataka - zatim generiše rezultate traženih podataka.
- Server web aplikacija šalje rezultate web serveru sa traženim informacijama ili obrađenim podacima.
- Web server klijentu odgovara uz tražene informacije koje se zatim pojavljuju na korisnikovom ekranu. Setite se onih vremena kada ste o putovanjima mogli da se informišete jedino fizičkim odlaskom do turističkih agencija ili prelistavanjem kataloga i časopisa. To vreme nije bilo davno. Tehnologija je u poslednje dve decenije eksponencijalno izmenila turističku industriju. Sa svim tehnološkim napretkom web mesta za onlajn rezervaciju

#### NAPOMENA:

**Ovaj rad proistekao je iz master rada čiji mentor je bio dr Srđan Sladojević, vanr. prof.**

putovanja pokazale su se kao blagoslov za savremeni svet. Možete da rezervišete karte, pronađete najbolje mesto za boravak i još mnogo toga.

U ovom radu će biti opisana izrada web aplikacije za turističku agenciju, od projektovanja baze podataka, korišćenih tehnologija do same implementacije koda u backend i frontend delu kao i kompletan deployment proces i kreiranje CI/CD pipeline-ova pomoću (besplatnog) Azure naloga.

U sistemu razlikujemo tri tipa korisnika: registrovani korisnik, agent, administrator. Sva tri tipa korisnika imaju različit stepen ovlašćenja. Registrovani korisnik (nakon što se uloguje) ima uvid u kompletne ponude destinacija, izbor apartmana i hotela za svaku destinaciju, kao i tip prevoza koji postoji za određene destinacije.

Ponude uključuju informacije, slike i cene za destinacije i smeštaj (apartmane i/ili hotele) kao i recenzije putnika. Odabir destinacije se vrši prvo na nivou države, zatim na nivou grada. Nakon toga se prikazuju smeštaji koji postoje u tom mestu, podeljeni na apartmane i hotele sa izlistanim detaljnim informacijama koje su neophodne svakom putniku koji planira buduće putovanje.

Pored toga, tu su i informacije o prevozu (sopstveni prevoz/autobus/avion) i shodno izboru prevoza tu su i potencijalne avionske takse, trajekt, prevoz od/do aerodroma... Takođe, registrovani korisnik može da napravi rezervaciju željenog aranžmana i ima istoriju svojih rezervacija koje se dele na potvrđene i nepotvrđene rezervacije.

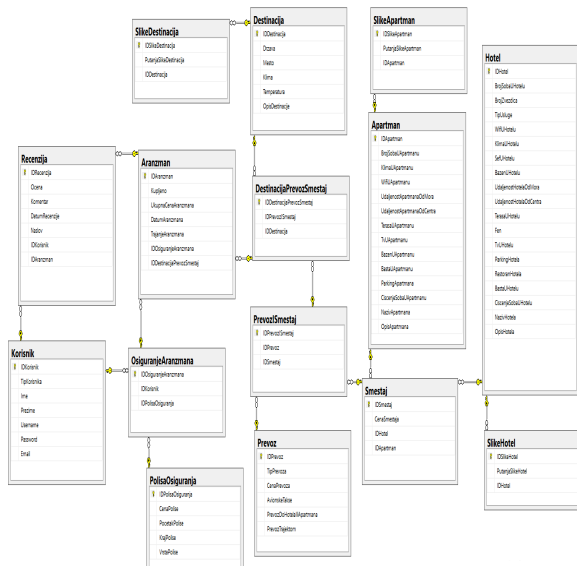
Rezervacije korisnika potvrđuje agent koji ima i mogućnost dodavanja, modifikovanja i brisanja, destinacija, hotela, apartmana i njihovih slika. Kada agent potvrdi rezervaciju, ona se kod korisnika prebacuje u deo potvrđenih rezervacija. Admin je na vrhu hijerarhije i ima sve permisije, dok korisnik koji nema kreiran nalog, bez logovanja, ima read-only prava i ograničen pristup aplikaciji.

#### 2. KORIŠĆENE TEHNOLOGIJE

Za izradu ovog projekta korišćene su različite tehnologije počevši od onih potrebnih za dizajniranje baze podataka, implementaciju backend dela do frontend dela same aplikacije kao i cloud tehnologije za deployment proces.

## 2.1. Baza podataka

Za izradu baze podataka korišćen je MS SQL Server u okviru Microsoft SQL Server Management Studio razvojnog okruženja. SQL Server je relacioni sistem upravljanja bazama podataka ili RDBMS, koji je razvio i plasirao Microsoft. Slično kao i drugi RDBMS softveri, SQL Server je izgrađen na vrhu SQL-a, standardnog programskog jezika za interakciju sa relacionim bazama podataka. SQL server je vezan za Transact-SQL ili T-SQL, Microsoft-ovu implementaciju SQL-a koja dodaje skup zaštićenih programskih konstrukcija [2].



Slika 1. Dijagram baze podataka

Smeštaj se deli na hotele i apartmane koji imaju svoje karakteristike kao i slike. Svaki smeštaj ima definisanu cenu. Prevoz ima definisan tip prevoza (sopstveni prevoz/avion/bus...), kao i opcione takse, prevoz do smeštaja, prevoz trajektom... Destinacija takođe ima svoje slike. Destinacija može imati više smeštaja i prevoza. Kao što i smeštaj može da ima više prevoza. Na primer, za destinaciju Grčka možemo imati više ponuđenih smeštaja (apartmana i/ili hotela) i za svaki taj pojedinačni smeštaj u ponudi se može naći i prevoz autobusom i avionom. Aranzman se sastoji od destinacije, smeštaja i prevoza, ukupne cene, datuma. Moguće je odabrati putno osiguranje izborom polise i unosom datuma. Korisnik može biti administrator, agent ili kupac. Takođe, korisnik može ostaviti recenziju i ocenu za neko putovanje na kom je bio.

## 2.2. Opis predloženog rešenja

Projekat predstavlja ASP .NET MVC aplikaciju. Arhitektura Model-View-Controller (MVC) deli aplikaciju na tri glavne komponente: model, view (prikaz) i kontroler. MVC framework uključuje sledeće komponente [3]:

- Model: Objekti modela su delovi aplikacije koji implementiraju logiku za domen podataka aplikacije. Objekti modela često preuzimaju i čuvaju stanje modela u bazi podataka. Ovi objekti modela preuzimaju i čuvaju stanje modela u bazi podataka.
- View: komponente koje prikazuju korisnički interfejs aplikacija. Ovi korisnički interfejsi se mogu

kreirati korišćenjem podataka o modelu, što može biti spisak ili pojedinačni objekat.

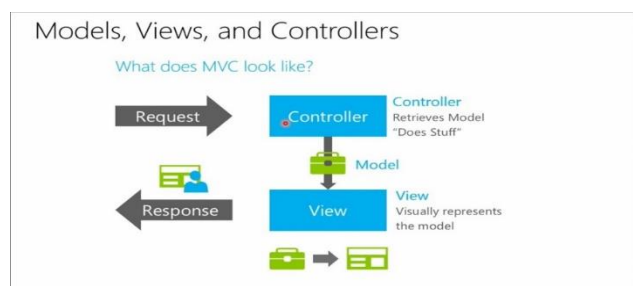
- Controller: komponente koje upravljaju poslovnom logikom i interakcijom korisnika. Oni takođe rade sa modelom i biraju prikaz, koji na kraju prikazuje UI. Kontroler šalje vrednosti primljene od korisnika (UI) modelu da bi preuzeo podatke iz baze podataka.

MVC obrazac vam pomaže da kreirate aplikacije koje odvajaju različite aspekte aplikacije (ulazna logika, poslovna logika i logika korisničkog interfejsa), istovremeno pružajući labavu spregu između ovih elemenata. Uzorak određuje gde treba da se nalazi svaka vrsta logike u aplikaciji. U pogledu pripada logika korisničkog interfejsa. Ulazna logika pripada kontroleru. Poslovna logika pripada modelu. Ovo razdvajanje vam pomaže da upravljate složenošću prilikom izrade aplikacije jer vam omogućava da se istovremeno fokusirate na jedan aspekt implementacije. Na primer, možete se usredsrediti na pogled bez zavisnosti od poslovne logike.

Aplikacija koristi Entity Framework 6 (EF6), koji je provereno i testirano objektno-relaciono mapiranje (O/RM) za .NET sa dugogodišnjim razvojem i stabilizacijom karakteristika [4].

Kao O/RM, EF6 smanjuje neusklađenost otpora između relacionog i objektno orijentisanog sveta, omogućavajući programerima da napišu aplikacije koje komuniciraju sa podacima skladištenim u relacionim bazama podataka. EF6 primenjuje mnoge popularne O/RM funkcije:

- Mapiranje klasa POCO entiteta
- Automatsko praćenje promena
- Rešavanje identiteta i jedinica rada
- Eager, lazy i explicit učitavanje
- Prevođenje upita pisanih pomoću LINQ-a (Language Integrated Query)
- Bogate mogućnosti mapiranja, uključujući podršku za:
  - One-to-one, one-to-many, many-to-many relacije
  - Nasleđivanje
  - Kompleksni tipovi
- Vizuelni dizajner za kreiranje entitetskih modela
- Integracija sa modelima aplikacija .NET Framework, uključujući ASP.NET, i putem vezivanja podataka, sa WPF i WinForms
- Povezivanje baza podataka zasnovano na ADO.NET-u i brojnim dobavljačima dostupnim za povezivanje na SQL Server, Oracle, MySQL, SQLite, PostgreSQL, DB2 itd.



Slika 2. MVC arhitektura

Za izradu frontend dela korišćeni su HTML, CSS i JavaScript. HTML pruža osnovnu strukturu web stranica, koju poboljšavaju i modifikuju druge tehnologije poput CSS-a i JavaScript-a. CSS se koristi za kontrolu prezentacije, formatiranja i izgleda. JavaScript se koristi za kontrolu ponašanja različitih elemenata.

### 2.3. Deployment

Microsoft Azure je Microsoft-ova javna platforma za računarstvo u cloud-u. Pruža niz usluga cloud usluga, uključujući računarstvo, analitiku, skladištenje i umrežavanje. Korisnici mogu da biraju između ovih usluga kako bi razvili i skalirali nove aplikacije ili pokrenuli postojeće aplikacije u javnom cloud-u. Kada se korisnici pretplate na Azure, imaju pristup svim uslugama uključenim u Azure portal. Pretplatnici mogu da koriste ove usluge za kreiranje resursa zasnovanih na cloud-u, kao što su virtuelne mašine (VM) i baze podataka.

Pošto se Microsoft Azure sastoji od brojnih usluga, slučajevi njegove upotrebe su izuzetno raznoliki. Pokretanje virtuelnih mašina ili kontejnera u cloud-u je jedna od najpopularnijih upotreba za Microsoft Azure. Ovi računarski resursi mogu da host-uju komponente infrastrukture, kao što su DNS serveri, Windows Server usluge - kao što su Internet Information Services (IIS) ili aplikacije trećih strana. Microsoft takođe podržava upotrebu operativnih sistema nezavisnih proizvođača, kao što je Linux.

Azure se takođe obično koristi kao platforma za host-ovanje baza podataka u cloud-u. Microsoft nudi relacione baze podataka bez servera kao što je Azure SQL i nerelacione baze podataka kao što je NoSQL.

Azure App Service je usluga zasnovana na HTTP-u za host-ovanje web aplikacija, REST API-ja i mobilnih backend-ova. Moguće je razvijanje na različitim jezicima, kao što su .NET, .NET Core, Java, Ruby, Node.js, PHP ili Python. Aplikacije se pokreću i skaliraju sa lakoćom u Windows i Linux okruženjima.

Azure App Service ne samo da dodaje snagu Microsoft Azure-a vašoj aplikaciji, kao što su bezbednost, balansiranje opterećenja, automatsko skaliranje i automatizovano upravljanje, već je moguće iskoristiti prednosti njegovih DevOps mogućnosti, kao što su CI/CD iz Azure DevOps, GitHub, Docker Hub i drugih izvora, upravljanje paketima, okruženja staging, prilagođeni domen i TLS/SSL sertifikati.

Uz App Service, plaćate za Azure računarske resurse koje koristite. Računski resursi koji se koriste određeni su planom usluge aplikacija na kojem se pokreću aplikacije.

Za potrebe deployment-a ASP.NET web aplikacije kreiran je Azure free account [5], zatim resource grupa i azure web app servis. Resource grupa je kontejner koji sadrži povezane resurse za Azure rešenje. Može uključivati sve resurse za rešenje ili samo one resurse kojima želimo da upravljamo [6]. Web apps su jedna od najčešće korišćenih Azure usluga koje nam omogućavaju da hostujemo naše web aplikacije u Azure-u. Može da hostuje korisničku interaktivnu aplikaciju ili čak pozadinsku uslugu kao što je WCF usluga ili web API-ji. Sa web apps možemo da iskoristimo snagu Microsoft

Azure-a i napravimo multifunkcionalne, izuzetno skalabilne, visoko bezbedne i besprekorno dostupne aplikacije zasnovane na internetu ili intranetu [7].

#### 2.3.1. Deployment lokalne baze na Azure

Pre deployment-a same aplikacije, bilo je potrebno odraditi deploy SQL baze na Azure koristeći Sql Server Management Studio (SSMS). Azure SQL baza podataka je inteligentna, skalabilna, relaciona usluga baze podataka kreirana za cloud. Predstavlja platformu kojom se u potpunosti upravlja kao servisom (PaaS → Platform as a Service) mehanizma baze podataka koji upravlja većinom funkcija upravljanja bazom podataka kao što su upgrade, patch, backup i monitoring bez učešća korisnika. Potrebno je kreirati Azure SQL database server i Azure SQL bazu nakon čega je moguće odraditi deploy lokalne baze na Azure.

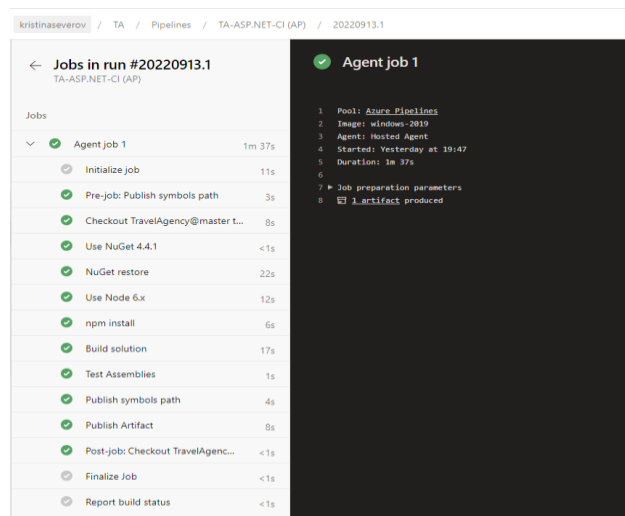
#### 2.3.2. Kreiranje CI/CD pipeline-a

Prvo je potrebno importovati projekat sa GitHub-a na Azure repozitorijum.

CI pipeline će izbildati source code, kreirati i nakon toga publish-ovati artefakte. Moguće je odabrati build template u zavisnosti od tipa projekta, u ovom slučaju je to ASP.NET. Odabrati source code konekciju, u našem slučaju to je Azure Repos Git pošto smo prethodno importovali projekat sa GitHub-a na Azure repozitorijum.

Nakon toga, potrebno je odabrati repozitorijum i granu. U okviru Process task-a popuniti build definiciju i podrazumevanog agenta, gde želimo da pokrenemo build. Nakon toga je potrebno odabrati i solution fajl našeg projekta. U Get Sources tasku je potrebno odabrati odgovarajući projekat i repozitorijum, a u Build Solution tasku biramo visual studio verziju i MSBuild arhitekturu.

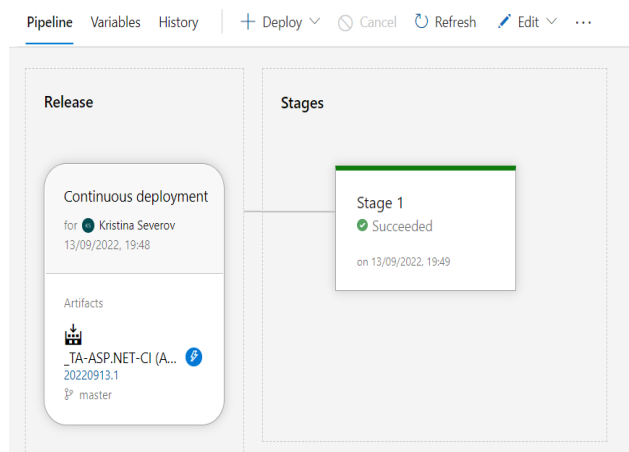
Moramo da dodamo npm task u build pipeline, kako bismo instalirali sve pakete koji se koriste u našem projektu. Node.js tool installer nam omogućava da pronađemo, preuzmemo i keširamo određenu verziju Node.js-a i dodamo je u putanju.



Slika 3. Uspešno izvršen CI pipeline

Korišćen je release pipeline, kako bismo pokupili artefakte koje je objavio naš build pipeline i zatim ih deploy na Azure web app. Prilikom kreiranja release pipeline-a je odabran Azure App Service Deployment jer želimo da odradimo deploy naše aplikacije na Azure app service.

Klikom na Add artifact se otvara novi prozor gde biramo projekat, build pipeline i default verziju (u našem slučaju uzima se uvek najnoviji build). Uključiti continuous deployment trigger i odabrati granu. U tasks tabu je potrebno odabrati odgovarajuću Azure pretplatu i autorizovati Azure pipeline-ove da se konektuju na Azure pretplatu i odabrati web app iz padajućeg menija, nakon čega je potrebno sačuvati naš release pipeline.



Slika 4. Uspešno izvršen release pipeline

Nakon što je release uspešno završen, potrebno je otvoriti URL naše web app koji se može naći na Azure portalu. Klikom na URL, u browser-u se otvara naša aplikacija.

### 3. ZAKLJUČAK

Ideja rada je bila da se prikaže proces razvoja web aplikacije za turističku agenciju:

- Prikaz dijagrama šeme baze podataka
- Baza podataka
- Backend
- Frontend deo projekta
- Deployment na web app service korišćenjem

CI/CD Azure pipeline-ova.

Opisane su funkcionalnosti, korisnici ovog sistema i način implementacije. Rezultat je user-friendly aplikacija koja korisnicima omogućava uvid u sve neophodne informacije prilikom rezervisanja željenog aranžmana, a agenti i admini na lak način mogu da manipulišu sa prikazom podataka i upravljanjem rezervacijama.

Aplikacija je primenjiva za različite turističke agencije sa različitim mogućnostima unapređenja kao što je uvođenje modula plaćanja koji bi sadržao listu svih plaćanja koje agencija obavlja. Tu bi se nalazile priznanice za sve uplate putnika i bankovni izvodi za sve isplate organizatorima putovanja.

Osim evidencije plaćanja moguć bi bio i pregled plaćanja i ostatka duga po rezervaciji (ugovoru).

Postavljeni su i CI/CD pipeline-ovi i naša aplikacija je deploy-ovana na Azure App Service, koji pruža skalabilnu uslugu web hostinga i olakšava komplikovanu proceduru deployment-a aplikacije. Korišćenjem Azure platforme, nude se mnogi benefiti od kreiranja do testiranja, deployment-a što omogućava bržu i efikasniju izradu aplikacija i na kompanijskom nivou.

### 4. LITERATURA

- [1] What is a Web Application? <https://blog.stackpath.com/web-application/>
- [2] What is MSSQL (Microsoft SQL)? <https://www.atlantic.net/what-is-mssql/>
- [3] ASP .NET MVC <https://medium.com/@theaxontech/asp-net-mvc-overview-a2ac7660539f>
- [4] Entity Framework <https://docs.microsoft.com/en-gb/ef/ef6/>
- [5] Create free Microsoft Azure account <https://k21academy.com/microsoft-azure/create-free-microsoft-azure-trial-account/>
- [6] Manage resource group <https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/manage-resource-groups-portal>
- [7] App service <https://azure.microsoft.com/en-us/services/app-service/>

### Kratka biografija:



**Kristina Ševerov** rođena je u Novom Sadu 1997. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Informaciono-komunikacionih sistema odbranila je 2020. god.  
kontakt:  
kristina.severov@gmail.com