

Implementacija Gatsby i Ghost CMS tehnologija pri izradi mrežnog mjesta

Pavić, Monika

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Humanities and Social Sciences / Sveučilište Josipa Jurja Strossmayera u Osijeku, Filozofski fakultet**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:142:029991>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-01-30**



Repository / Repozitorij:

[FFOS-repository - Repository of the Faculty of Humanities and Social Sciences Osijek](#)



Sveučilište J.J. Strossmayera u Osijeku

Filozofski fakultet Osijek

Dvopredmetni diplomski studij nakladništva i informacijskih tehnologija

Monika Pavić

IMPLEMENTACIJA GATSBY I GHOST CMS TEHNOLOGIJA PRI IZRADI
MREŽNOG MJESTA

Diplomski rad

Mentor: izv. prof. dr. sc. Tomislav Jakopec

Osijek, 2024.

Sveučilište J.J. Strossmayera u Osijeku

Filozofski fakultet Osijek

Odsjek za informacijske znanosti

Dvopredmetni diplomski studij nakladništva i informacijskih tehnologija

Monika Pavić

IMPLEMENTACIJA GATSBY I GHOST CMS TEHNOLOGIJA PRI IZRADI
MREŽNOG MJESTA

Diplomski rad

Društvene znanosti, Informacijske i komunikacijske znanosti, Informacijski sustavi
i informatologija

Mentor: izv. prof. dr. sc. Tomislav Jakopec

Osijek, 2024.

IZJAVA

Izjavljujem s punom materijalnom i moralnom odgovornošću da sam ovaj rad samostalno napravio te da u njemu nema kopiranih ili prepisanih dijelova teksta tuđih radova, a da nisu označeni kao citati s napisanim izvorom odakle su preneseni. Svojim vlastoručnim potpisom potvrđujem da sam suglasan da Filozofski fakultet Osijek trajno pohrani i javno objavi ovaj moj rad u internetskoj bazi završnih i diplomskih radova knjižnice Filozofskog fakulteta Osijek, knjižnice Sveučilišta Josipa Jurja Strossmayera u Osijeku i Nacionalne i sveučilišne knjižnice u Zagrebu.

U Osijeku, 4.9.2024.

Pačić Monika, 0122225999

SAŽETAK

Cilj ovog diplomskog rada je definirati CMS tehnologiju kao što su Gatsby i Ghost te prikazati njihovu implementaciju pri izradi mrežnog mjesta. Rad objašnjava povijest sustava za upravljanje sadržajem te kako se on mijenjao i razvijao kroz godine uz detaljnije pojašnjenje i razvoja samog weba kao i tehnologija povezanih sa sustavima za upravljanje sadržajem. Nadalje, pojašnjava pojedine tehnologije Gatsby i Ghost te koja je njihova uloga u razvoju mrežnog mjesta. Nakon toga prikazuje se kako preko Gatsby *frontend* dijela i Ghost API-ja nastaje razvoj mrežnog mjesta te je prikazan primjer Gatsby mrežnog mjesta, odnosno jednostavan primjer *hello world* i CRUD (*create, read, update, delete*) aplikacija. Prikazanim primjerima daje se jasnije pojašnjenje na koji način CMS tehnologija funkcionira te na koji način olakšava razvojnim programerima razvoj mrežnih mjesta kao i na koji način navedene tehnologije funkcioniraju zajedno. U raspravi su navedene prednosti i nedostaci kao i problematika koja se pojavila prilikom kreiranja aplikacije. Na kraju, zaključak objedinjuje sve što je u radu napisano te pojašnjava što za razvoj mrežnog mjesta znači koristiti sustave za upravljanje sadržajem.

KLJUČNE RIJEČI: cms tehnologija, Ghost, Gatsby, mrežno mjesto

Sadržaj

1.	UVOD.....	1
2.	SUSTAV ZA UPRAVLJANJE SADRŽAJEM	3
2.1.	O sustavu za upravljanje sadržajem	3
2.2.	Povijest sustava za upravljanje sadržajem	3
3.	CMS TEHNOLOGIJA	13
4.	GATSBY.....	17
5.	GHOST	22
6.	GATSBY + GHOST.....	26
7.	APLIKACIJA	39
8.	RASPRAVA	50
9.	ZAKLJUČAK	53
10.	LITERATURA	54

1. UVOD

Razvojem tehnologije i digitalnog svijeta, danas je jedan od ključnih elemenata i CMS tehnologija. CMS tehnologija (sustav za upravljanje sadržajem) postaje važan element u digitalnom svijetu; omogućuje lako kreiranje, uređivanje i upravljanje mrežnim stranicama. Sustav je pristupačan i svestran te neophodan za prisutnost na Internetu, kako poduzetnika i korporacija, tako i blogera¹. U radu je objašnjeno na koji način CMS tehnologija pridonosi digitalnom svijetu, koje su njezine prednosti i kako pojednostavljuje izradu mrežnih mjesta i stranica. Kao primjere CMS sustava opisani su i prikazani Ghost te njegova implementacija s Gatsby *frontend* okvirom te kreiranje jednostavnog mrežnog mjesta. Ghost je CMS sustav za profesionalne izdavače i blogere koji je namijenjen za kreiranje, dijeljenje i razvoj poslovanja koristeći sadržaj. Koristi moderne alate za izradu mrežnih stranica, objavljivanje sadržaja i sl.² Gatsby je JavaScript *frontend* okvir za izradu mrežnih stranica temeljen na Reactu; pruža ugrađenu sigurnost, skalabilnost te je fokus na izvedbi.³ Kako bi se omogućio razvoj mrežnog mjesta pomoću Gatsby i Ghost CMS-a potrebno je kao temelj imati instaliran paket Nodejs i node package manager. Node.js je besplatno JavaScript okruženje otvorenog koda koje omogućuje stvaranje poslužitelja, mrežnih aplikacija, alata naredbenog retka i skripti.⁴ Nadalje, Node package manager, skraćeno npm, je standardni upravitelj paketa za Node.js.⁵

Rad prvo objašnjava povijest i razvoj tehnologije i kako je to dovelo do razvoja CMS sustava, nakon čega se definira i objašnjava rad CMS tehnologije. Također, objašnjava i definira Gatsby i Ghost sustave nakon čega slijedi praktični prikaz implementacije Ghost i Gatsby tehnologije pri izradi mrežnog mjesta. Rasprava i zaključak povezuju sve što je u prethodnim poglavljima

¹ Usp. Što je CMS? Pregled sustava za upravljanje sadržajem i njihova primjena u izradi web stranica, 2024. URL: <https://copymate.app/hr/blog/multi/sto-je-cms-pregled-sustava-za-upravljanje-sadrzajem-i-njihova-primjena-u-izradi-web-stranica/> (2024-08-10)

² Usp. Ghost. Independent technology for modern publishing. URL: <https://ghost.org/> (2024-08-10)

³ Usp. Gatsby. Welcome to the Gatsby Way of Building. URL: <https://www.gatsbyjs.com/docs/> (2024-08-10)

⁴ Usp. Nodejs. Run JavaScript Everywhere. URL: <https://nodejs.org/en> (2024-08-10)

⁵ Usp. Nodejs. An introduction to the npm package manager. URL: <https://nodejs.org/en/learn/getting-started/an-introduction-to-the-npm-package-manager> (2024-08-10)

navedeno, kako teorijski tako i praktični dio te prati li izrada aplikacije i implementacija Gatsby i Ghost tehnologije ono što je u teoriji opisano. Također, navedene su prednosti i nedostaci korištenja istih kao i izazovi i problemi na koje se naišlo prilikom kreiranja CRUD aplikacije.

2. SUSTAV ZA UPRAVLJANJE SADRŽAJEM

2.1. O sustavu za upravljanje sadržajem

Sustavi za upravljanje sadržajem (CMS) važan su dio stvaranja i upravljanja digitalnim sadržajem. Pružaju alate i značajke koje su potrebne za organizaciju izrade mrežne stranice, internetsku trgovinu ili upravljanje bilo kakvim digitalnim sadržajem.⁶

2.2. Povijest sustava za upravljanje sadržajem

Web 1.0

Prije definiranja i objašnjenja funkcionalnosti sustava za upravljanje sadržajem, potrebno je objasniti kako je došlo do njihovog razvoja.

Web 1.0 izraz je koji označava prvu fazu razvoja World Wide Weba koju karakteriziraju jednostavne statične mrežne stranice. Povijest sustava za upravljanje sadržajem kreće 1989. godine kada Tim Berners-Lee predlaže internetski hipertekstualni sustav HTML te nastaje preglednik i poslužiteljski softver 1990. godine. HTML nastaje iz SGML-a (Standard Generalized Markup Language) a kreirali su ga u IBM-u Charles F. Goldfarb, Ed Mosher i Ray Lorie 1970-ih godina. Prve mrežne stranice bile su jednostavne HTML tekstualne datoteke. Korišten je FTP program za kopiranje datoteka u direktorij pod pokrenutim mrežnim poslužiteljem. 1993. godine preglednici su počeli podržavati slike koje su se pojavljivale s tekстом, a statične stranice koje su izgledale kao brošure sadržavale su informacije o tvrtkama i proizvodima.⁷

Prvi korak u upravljanju sadržajem na mrežnim stranicama počeo je 1990-ih godina sa *Server Side Includes* (SSI). SSI omogućuje da se pojedini dijelovi mrežnog mjesta odvoje od glavnog sadržaja,

⁶Usp. History of content management systems and rise of headless CMS, 2023. URL: <https://www.contentstack.com/blog/all-about-headless/content-management-systems-history-and-headless-cms> (2024-05-19)

⁷ Usp. Isto.

kao što su to izbornik ili podnožje. Tada se pojavljuje i *Common Gateway Interface* koje omogućuje stvaranje interaktivnih obrazaca.

Prema Tim Berner – Leeju cilj HTML-a je odvajanje strukture dokumenta od izgleda dokumenta. 90-ih godina nije bilo moguće stilizirati dokumente, npr. prikaz novinskog izgleda s više stupaca na mrežnoj stranici. Potrebno je bilo izgraditi stilski jezik pa je tako do 1995. godine *Wide Web Consortium* (W3C) bio u upotrebi.

Prvi komercijalni preglednik koji podržava CSS je Microsoftov Internet Explorer 3 1996. godine. Nakon njega Netscape Communicator, verzija 4.0.

ColdFusion 1996. godine dodaje skriptni jezik CFML. U razdoblju od 1995. do 1997. godine vladalo je skriptiranje na strani poslužitelja. Za vrijeme tog razdoblja, Personal Home Page (PHP) i Active Server Pages (ASP) počinju skriptiranje na strani poslužitelja za generiranje sadržaja poslanog s poslužitelja na mrežni preglednik. Također, 1999. godine pojavljuje se i JavaServer Pages (JSP) koji je izgrađen oko programskog jezika Java.

Microsoft 1997. godine uvodi *iframe* koji omogućuju dijeljenje HTML prozora preglednika u segmente gdje svaki okvir prikazuje drugačiji dokument koji se može koristiti za prikaz sadržaja s drugih stranica te je bio popularan za predstavljanje oglasa i *banner*.⁸

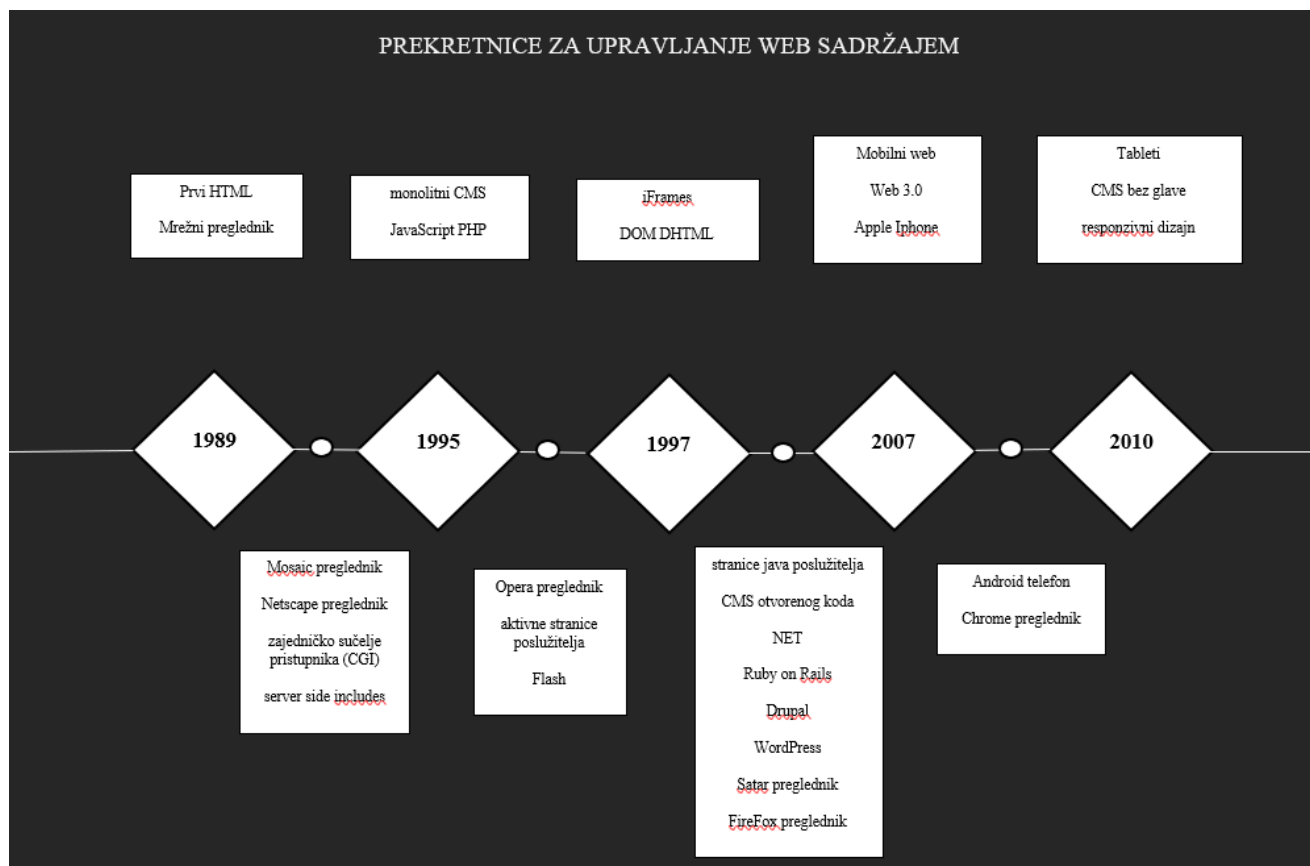
DOM i dinamička HTML revolucija

Dinamički sadržaj pojavljuje se 1997. godine uvođenjem Document Object Model (DOM). DOM definira logičku strukturu dokumenata koja omogućuje prepoznavanje i programsku kontrolu dijelova dokumenta. DOM je sučelje za programiranje aplikacija (API) za HTML i XML dokumente; omogućuje pristup i manipuliranje stilovima HTML elemenata.

Ajax, odnosno dinamički HTML koji koristi asinkroni JavaScript i XML omogućuje razvojnim programerima da traže i primaju podatke za ažuriranje mrežne stranice bez ponovnog učitavanja stranice.⁹

⁸ Usp. Isto.

⁹ Usp. Isto.



Slika 1. Web 1.0¹⁰

Slika 1. prikazuje prekretnice i neke od glavnih događaja koji su važni za razvoj mrežnog sadržaja, počevši od prvog HTML-a 1989. godine do 2010. godine kad se pojavljuje prilagodljivi dizajn koji se prilagođava zaslonima različitih veličina.

Web 2.0 i uloga CMS-a

Dinamička struktura sadržaja donosi drugačiji pristup interakcije s mrežnim sadržajem, naglašava da su stranice društvene. Web 2.0 odnosi se na sadržaj koji korisnici generiraju te jednostavnost korištenja kako bi mrežne stranice funkcionirale s drugim sustavima i proizvodima.

¹⁰ Usp. Isto.

Prijelaz sa statičnih stranica na dinamičke stranice rezultirao je potrebom za upravljanjem sadržajem; potrebno je svakodnevno ažuriranje mrežnih stranica a postoje različite uloge kako bi se to izvršilo. Tako marketinški odjel upravlja promotivnim materijalom, Ljudski potencijali bave se objavom novih poslova, Odjel za odnose s javnošću objavljuje priopćenje za javnost, Odjel za dokumente objavljuje dokumentaciju proizvoda, odjel podrške komunicira s klijentima na mreži i sl. Sama uloga sustava za upravljanje sadržajem je pružiti mogućnost više korisnika na različitim razinama upravljanje sadržajem ili dijelom sadržaja na mrežnom mjestu.¹¹

Osnovne komponente CMS-a

Sustav za upravljanje sadržajem čine dva temeljna elementa, aplikacija za upravljanje sadržajem (CMA) i aplikacija za isporuku sadržaja (CDA). CMA omogućuje administraciju korisnika i grupa kako bi se sadržaj na mrežnoj stranici mogao stvarati, uređivati i uklanjati. Uključuje i *frontend* korisničko sučelje koje omogućuje dodavanje, izmjenu i uklanjanje sadržaja s mrežnog mjesta bez potrebnog poznavanja HTML-a, CSS-a i programskih jezika. CDA prikuplja te informacije i ažurira mrežnu stranicu.¹²

Uspon monolitnog CMS-a

Monolitni CMS je spojeni sustav, odnosno sveobuhvatno rješenje za upravljanje sadržajem te uključuje sve što je potrebno za upravljanje i objavljivanje sadržaja na mreži.

FileNet, osnovan 1985. godine, smatra se prvim sustavom koji je prirodan sustav za upravljanje sadržajem. 1995. godine predstavlja kompletan paket programa za integrirano upravljanje dokumentima. Iste, 1995. godine osnovan je Vignette koji objavljivanje na mreži čini pristupačnijim i personaliziranijim te prema njemu nastaje pojam 'sustav za upravljanje sadržajem'. Nakon toga, 90-ih godina, pojavljuju se brojni poslovni CMS-ovi kao što su Interwoven, Documentum, FatWire, FutureTense, Inso, EPiServer.¹³

¹¹ Usp. Isto.

¹² Usp. Isto.

¹³ Usp. Isto.

Open-source CMS i okviri

Nadalje, 2000-ih godina pojavljuju se i okviri za upravljanje sadržajem otvorenog koda. Okvir (*framework*) je programska biblioteka unaprijed napisanog koda. Tada se pojavljuju i besplatne alternative, tj. Sustavi za upravljanje sadržajem kao što su OpenCMS, PHP Nuke, Mambo, WordPress, Drupal, Plone i Joomla. WordPress postaje jedan od najpopularnijih CMS sustava jer daje rješenje otvorenog koda koje je fokusirano na isporuku sadržaja i dopušta programerima da dodaju prilagodbe i proširenja. 2006. godine nastaje alternativa otvorenog koda za upravljanje poslovnog sadržaja; Alfresco.¹⁴

Porast platforme za izradu mrežnih stranica

Početak 2000-ih godina nastaju unaprijed pripremljeni predlošci koji nisu zahtijevali iskustvo u kodiranju, odnosno poznavanje HTML-a i CSS-a. To su WordPress, Squarespace, Weebly i Wix.

Web API-ji, XML i JSON

Dio Web 2.0. bio je i rad mrežnih stranica s drugim proizvodima i sustavima. Mrežni API je programsko sučelje koje omogućuje pristup mrežnim stranicama putem standardnih metoda HTTP zahtjeva. Podaci su obično u XML ili JSON formatu.

XML (*extensible Markup Language*) dolazi od SGML-a (*Standard Generalized Markup Language*), kao i HTML. XML omogućuje prijenos podataka putem API poziva jer je format neovisan o platformi.

JSON (*JavaScript Object Notation*) je format za pohranjivanje serijaliziranih podataka s parovima ključ-vrijednost i prijenos tih podataka između poslužitelja i mrežne aplikacije. JSON *feedovi* učitavaju se brže od XML i RSS *feedova*; ima prednost nad XML-om jer ima manji otisak, lakši je za korištenje i izvrsno radi s preglednicima koji imaju omogućen JavaScript s obzirom da JavaScript automatski prepoznaje JSON.¹⁵

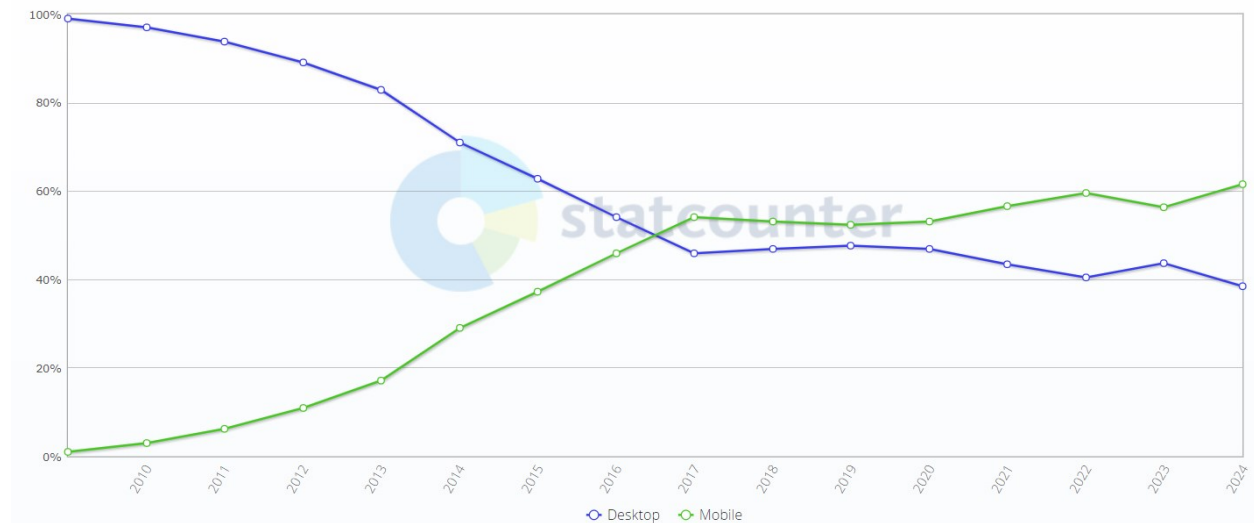
¹⁴ Usp. Isto.

¹⁵ Usp. Isto.

Web 3.0.

Krajem 1990-ih i početkom 2000-ih godina neki mobilni uređaji omogućavali su pristup mreži, ali tek uvođenjem iPhone-a 2007. godine i Android pametnog telefona 2008. godine imaju utjecaj na isporuku sadržaja na mreži. Pružanje sadržaja na mobilnim uređajima naziva se Web 3.0 kako bi se označio prelazak s računala i prijenosnih računala na mobitele.

S obzirom da je korištenje mobilnog interneta premašilo korištenje stolnog interneta, javlja se problem za monolitni CMS koji isključivo isporučuje sadržaj stolnim i prijenosnim računalima. Iz tog razloga, počele su se kreirati mobilne i *desktop* verzije mrežnih stranica s mobilnim dizajnom koji pruža skraćene verzije mrežnih stranica pa je tako od 2009. godine do 2024. udio korištenja mobilnih verzija narastao za 60% u odnosu na *desktop* verziju gdje je udio pao sa 100% na ispod 40% (slika 2.).



Slika 2. Web 3.0¹⁶

¹⁶ Usp. Statcounter. Mobile market share worldwide. URL: <https://gs.statcounter.com/platform-market-share/desktop-mobile/worldwide/#yearly-2009-2024> (2022-09-16)

Mobilne mrežne lokacije bile su na zasebnoj poddomeni pod nazivom mobilna ili m.dot. no, Google ne osigurava indeksiranje m.dot stranica nego ih samo označava kako bi se označilo da je mrežno mjesto prilagođeno mobilnom uređaju.

Kao posljedica toga, 2010. godine nastaje dizajn koji se prilagođava različitim širinama zaslona koji promovira pomak mrežnih stranica na prilagodljivost različitim širinama zaslona i fluidne izgleda. W3C tako razvija mehanizam upita za identifikaciju širine zaslona (*media query*) kao dio CSS3 specifikacije. Medijski upit omogućuje utvrđivanje vrste uređaja i pregled fizičkih karakteristika uređaja (veličina zaslona).¹⁷

Promjena paradigme na višekanalnost

Pojavom novih kanala; pametni satovi, igraće konzole, uređaji s glasovnom aktivacijom poput *Alexe* i *Google Home* nastaje problem s isporukom sadržaja za tradicionalni CMS. Promjenom paradigme, isporuka sadržaja za nekoliko kanala do dostave višekanalnog sadržaja, nastaje razdvojeno i bezglavo CMS rješenje.¹⁸

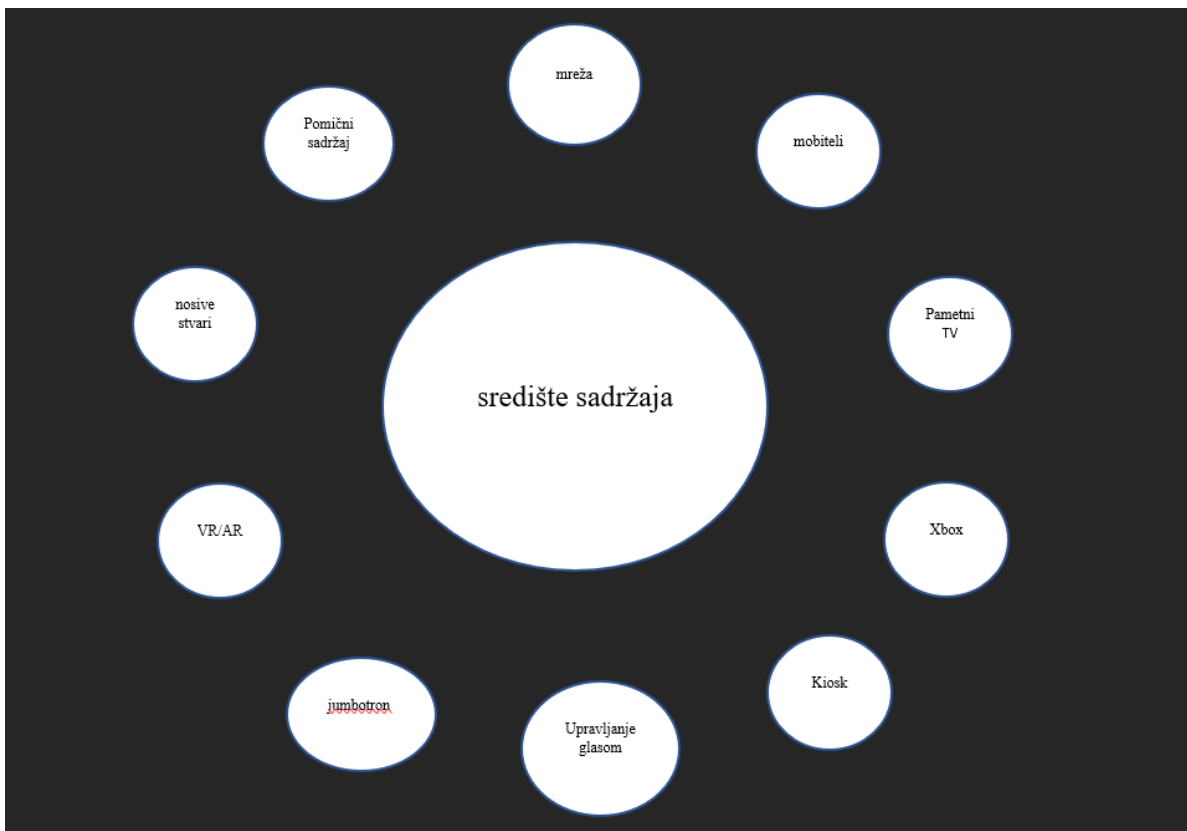
Arhitektura središta sadržaja

Središte sadržaja centralizira sav sadržaj na jednom mjestu pomoću API-ja za isporuku sadržaja bilo gdje. Takav pristup ubrzava i pojednostavljuje upravljanje sadržajem te omogućava korištenje najboljih alata za stvaranje platformi za digitalno iskustvo (DXP) s višekanalnom isporukom sadržaja (slika 3.).¹⁹

¹⁷ Usp. Isto.

¹⁸ Usp. Isto.

¹⁹ Usp. Isto.



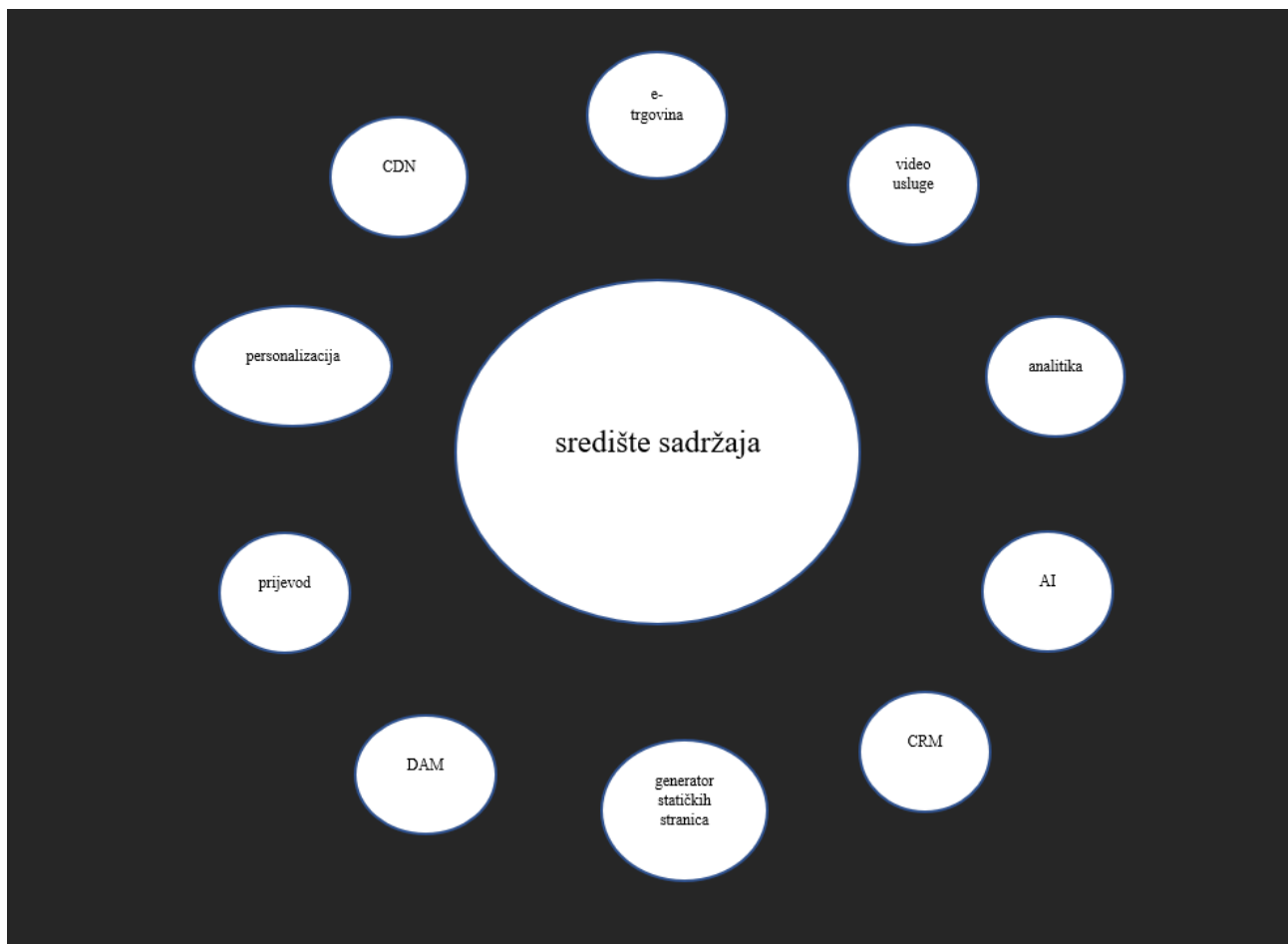
Slika 3. Arhitektura sadržaja ²⁰

Integracija i okviri

Arhitektura središta sadržaja omogućuje i više slobode za integraciju. Bezglavi CMS omogućava odabir najboljih alata i usluga, poput alata za automatizaciju marketinga, analitika, mehanizam za personalizaciju, usluge prevođenja itd. (slika 4.)²¹

²⁰ Usp. History of content management systems and rise of headless CMS, 2023. URL: <https://www.contentstack.com/blog/all-about-headless/content-management-systems-history-and-headless-cms> (2024-05-19)

²¹ Usp. Isto.



Slika 4. Arhitektura sadržaja²²

Sigurnosna i CMS rješenja

Većina bezglavih CMS sustava je u kategoriji sadržaja kao usluge (CaaS); usluga je centralizirana i smještena u oblaku što olakšava skaliranje.

Tehnologija koja također utječe na isporuku sadržaja na globalnoj razini je CDN, mreža za isporuku sadržaja. Statička imovina i dinamički sadržaj mrežne stranice predmemorira se i sprema

²² Usp. Isto.

na svim poslužiteljima CDN-a. Kad se zatraži stranica, mrežna stranica dohvaća predmemorirani sadržaj s najbližeg CDN poslužitelja i isporučuje ga klijentu.

Bezglavi CMS je ključna komponenta nove generacije platformi za digitalno iskustvo (DXP).²³

Važnost personalizacije

Personalizacija je ključna za izgradnju učinkovite platforme digitalnog iskustva. To je razumijevanje interesa korisnika i prilagođavanje sadržaja koji odgovara različitim potrebama. Korištenjem bezglavog CMS-a, osobni podaci su dostupni putem API-ja, mrežnih usluga i standarda otvorenih podataka.

Neki od alata i usluga za personalizaciju su Optimizely²⁴, Monetate²⁵, One Spot²⁶, Evergage²⁷, Salesforce Commerce Cloud²⁸ i Adobe Target²⁹. Oni utječu na ponašanje posjetitelja, lokaciju i profil kako bi se stvorilo dinamičko personalizirano, relevantno iskustvo.³⁰

²³ Usp. Isto.

²⁴ Usp. Optimizely. URL: <https://www.optimizely.com/> (2024-09-09)

²⁵ Usp. Monetate. URL: <https://monetate.com/> (2024-09-09)

²⁶ Usp. One Spot. URL: <https://www.onespotapps.com/> (2024-09-09)

²⁷ Usp. Evergage. URL: <https://www.salesforce.com/marketing/personalization/> (2024-09-09)

²⁸ Usp. Salesforce Commerce Cloud. URL: <https://www.salesforce.com/commerce/> (2024-09-09)

²⁹ Usp. Adobe Target. URL: <https://business.adobe.com/products/target/adobe-target.html> (2024-09-09)

³⁰ Usp. History of content management systems and rise of headless CMS, 2023. URL: <https://www.contentstack.com/blog/all-about-headless/content-management-systems-history-and-headless-cms> (2024-05-19)

3. CMS TEHNOLOGIJA

Content management system, skraćeno CMS, odnosno Sustav za upravljanje sadržajem je program za stvaranje, uređivanje i organiziranje digitalnog sadržaja. Mjesto je za pohranu sadržaja i automatizaciju procesa zajedničkog upravljanja određenog sadržaja. Uključuje upravljanje sadržajem na mrežnom mjestu kao i upravljanje dokumentima, slikama, videima i ostalim vrstama sadržaja pomoću kojih se komunicira s korisnicima. Također, koristi se za konfiguraciju postavki mrežnih stranica te omogućuje stvaranje objava ali i izmjenu dizajna mrežnih stranica. Dodatne značajke sustava za upravljanje sadržajem su funkcije pretraživanja, optimizacija tražilice, sigurnost i sl., a dostupne su preko proširenja, tj. programskih komponenti gdje su dostupne dodatne funkcionalnosti koje ne postoje na originalnoj platformi.

Neke od prednosti CMS tehnologija su bolja suradnja i organizacija, odnosno sustav upravljanjem tijekom rada i pohranom sadržaja pomaže pri organizaciji istog pa je moguće kreiranje i objavljivanje sadržaja. Pristup je dostupan svima pa nema potrebe za višestrukim slanjem datoteka što smanjuje broj pogrešaka. Nadalje, CMS tehnologije imaju ugrađene korisničke uloge i dopuštenja što omogućuje siguran i učinkovit rad. Poslovi se mogu podijeliti na kreiranje sadržaja i upravljanje sadržajem. Kod kreiranja sadržaja korisnici imaju pristup alatima za pisanje, uređivanje i objavljivanje digitalnog sadržaja, a kod upravljanja sadržajem korisnici mogu brisati isti i mijenjati funkcionalnosti same stranice. Također, CMS tehnologija omogućuje dodavanje proširenja za bolje rezultate pretraživanja, odnosno omogućuje poboljšanje optimizacije za tražilice (SEO) te povećanje korištenja mrežne stranice. To su proširenja za analizu stranica, poboljšanje kompatibilnosti i prilagodljivosti različitim širinama sadržaja, prognoziranje i analiza ključnih riječi i sl. S obzirom na dostupnost spomenutih proširenja, korisnicima ostaje više vremena za izradu kvalitetnog sadržaja i bolju komunikaciju unutar organizacije.³¹

CMS ima značajnu ulogu u uspostavljanju i upravljanju internetskim trgovinama. CMS platforme pružaju alate za popis proizvoda, upravljanje zalihama, obradu plaćanja i obradu upita za korisničku službu. Neki CMS sustavi dizajnirani su za e-trgovinu, kao što su Shopify i Magento,

³¹ Usp. Što je CMS. URL: <https://www.virtualna-tvornica.com/sto-je-cms/> (2024-05-19)

dok se drugi sustavi mogu integrirati s platformama za internetsku trgovinu za olakšavanje transakcija. Integracija CMS sustava s platformama za internetsku trgovinu osigurava da je aplikacija za isporuku sadržaja (CDA) sinkronizirana s aplikacijom za upravljanje sadržajem (CMA).

CMS sustavi također upravljaju i digitalnom imovinom; upravitelj digitalne imovine integriran s CMS sustavom može pomoći tvrtkama u upravljanju različitim vrstama digitalnog sadržaja (slike, videozapisi i druge multimedijske datoteke), mogu se jednostavno organizirati, pristupiti im i ponovno ih koristiti.³²

Kada govorimo o samom funkcioniranju CMS-a, on radi na način da omogućuje pristup bazi podataka mrežne stranice putem grafičkog korisničkog sučelja, preko mrežnog preglednika. Unutar samog sučelja moguće je pristupiti alatima za upravljanje sadržajem koji omogućuju stvaranje i objavljivanje novih povezanih mrežnih stranica te ažuriranje postojećeg sadržaja. Postoje unaprijed postavljeni predlošci i kategorije za organizaciju mrežne stranice koji omogućuju vizualnu dosljednost na povezanim mrežnim stranicama. Omogućeno je upravljanje strukturama i navigacijom same mrežne stranice te pohranjivanje različitog sadržaja u baze podataka.

Cilj sustava za upravljanje sadržajem je pomaganje korisnicima upravljati podacima; funkcije sustava uključuju pohranjivanje, upravljanje, traženje, indeksiranje, kontrolu, objavljivanje te izvještavanje. Neke od dodatnih funkcija i značajki su prilagodljivi predlošci za upravljanje prezentacijom sadržaja, jezična i korisnička podrška, čarobnjak za instalaciju i nadogradnju, upravljanje verzijama, hijerarhija sadržaja, platforma za suradnju i sl. Također, može služiti i za marketinške kampanje putem *weba* te za unapređenje odnosa s klijentima.³³

³² Usp. History of content management systems and rise of headless CMS, 2023. URL: <https://www.contentstack.com/blog/all-about-headless/content-management-systems-history-and-headless-cms> (2024-05-19)

³³Usp. Što je CMS. URL: <https://www.virtualna-tvornica.com/sto-je-cms/> (2024-05-19)

Tri popularna sustava za upravljanje sadržajem

WordPress

WordPress je CMS koji je poznat po svojoj jednostavnosti i lakoći korištenja. Sastoji se od bogate biblioteke dodataka pomoću koje poboljšava funkcionalnost mrežne stranice. No, ima ograničenu skalabilnost za složene mrežne stranice i može biti sigurnosno ranjiv. Stoga, WordPress najbolje odgovara malim i srednjim tvrtkama, blogerima i pojedincima.

Joomla

Joomla je fleksibilan i prilagodljiv CMS sustav; ističe se u upravljanju mrežnih stranica e-trgovina. Podržava više jezika, ali ima manje dodataka od WordPressa te strmiu krivulju učenja. Idealan je CMS sustav za male i srednje tvrtke i mrežne stranice za e-trgovinu.

Drupal

Drupal je CMS sustav otvorenog koda i idealan je za složene mrežne stranice. Skalabilan je i pruža izdržljive i kompaktne sigurnosne značajke. Zahtijeva tehničku stručnost te također ima strmiu krivulju učenja. Pogodan je za velika poduzeća, vladine organizacije i obrazovne ustanove.³⁴

Bezglavi CMS

Headless CMS je sustav za upravljanje sadržajem koji omogućuje kreiranje i organiziranje sadržaja ali i razvojnim programerima pruža strukturirane podatke koji se prikazuju pomoću zasebnog sustava na sučelju aplikacije ili mrežnog mjesta. Za razliku od tradicionalnog CMS-a, koji se bavi pozadinskim upravljanjem sadržaja i poslužuje ga krajnjim korisnicima, *Headless CMS* nije povezan sa *frontend* dijelom sučelja što razvojnim programerima omogućuje razvoj iskustava krajnjim korisnicima koristeći najbolju tehnologiju koja je dostupna. Jednostavno

³⁴ Usp. History of content management systems and rise of headless CMS, 2023. URL: <https://www.contentstack.com/blog/all-about-headless/content-management-systems-history-and-headless-cms> (2024-05-19)

rečeno, upravlja sadržajem na način da se sadržaj pohranjuje nevezano za *frontend*, što ga čini dostupnim na svim uređajima i platformama.³⁵

Razdvojeno i bezglavo CMS rješenje

Razdvojeni sustav se sastoji od dva ili više sustava koji obavljaju transakcije bez povezivanja. Omogućuje promjenu sloja prezentacije (formatiranje) i ponašanja (programiranje) bez utjecaja na sadržaj stranice.

Iako su razdvojeni i bezglavi CMS često korišteni kao sinonimi, postoji razlika. Bezglavi CMS nema *frontend* sustav, integrira alate za upravljanje sadržaja putem API-ja. Odvojeni CMS uključuje *frontend*.³⁶

³⁵Usp. Gatsby. How to Source Content from a Headless CMS. URL: <https://www.gatsbyjs.com/docs/how-to/sourcing-data/headless-cms/> (2024-05-19)

³⁶ Usp. History of content management systems and rise of headless CMS, 2023. URL: <https://www.contentstack.com/blog/all-about-headless/content-management-systems-history-and-headless-cms> (2024-05-19)

4. GATSBY

Gatsby je *frontend* okvir otvorenog koda izgrađen na Node.js- u temeljen na React-u s ugrađenim performansama, skalabilnošću i sigurnošću. Omogućuje preko 2500 dodataka za stvaranje statičkih mrežnih stranica³⁷. Mnogi tradicionalni CMS-ovi podržavaju razdvojeni način rada što omogućuje razvoj Gatsby stranice. Neki od sustava za upravljanje sadržajem koji podržavaju su Wordpress i Drupal. Gatsby podržava desetke *headless CMS* opcija putem korištenja izvornih dodataka te time omogućuje kreatorima sadržaja prepoznavanje preferiranog *admin* sučelja.

Neki od sustava za upravljanje sadržajem koji su popularni kod Gatsby korisnika te podržavaju ključne funkcije pregleda sadržaja i inkrementalne izgradnje su Contentful, WordPress, Shopify, Sanity, Strapi, Contentstack, DatoCMS, Drupal, Prismic, NetlifyCMS.

Neki sustavi za upravljanje sadržajem mogu se pronaći u dodatcima; ako je riječ o CMS-u koji je integriran s Gatsbyjem može se dodati putem zahtjeva za dodavanje, a ako CMS nije integriran s Gatsbyjem, postoji mogućnost pisanja vlastitog izvornog dodatka.³⁸

Kada govorimo o karakteristikama Gatsbyja, možemo spomenuti *Gatsby framework*, Gatsby podatkovni sloj i *Gatsby Cloud*. Gatsby Framework omogućuje korištenje okvira temeljenih na React-u koji su otvorenog izvora. Gatsby podatkovni sloj omogućuje integraciju različitog sadržaja, API-a i usluga u jedinstveno mrežno iskustvo kao i kombiniranje podataka iz različitih izvora te prikazivanje istih jedne uz druge. *Gatsby Cloud* omogućuje izgradnju, pretpregled i raspored mrežnog mjesta u stvarnom vremenu.

Gatsby kombinira generiranje statičke mrežne lokacije, odgođeno generiranje statičke stranice i inteligentno prikazivanje stranice za selektivno prethodno učitavanje važnog sadržaja te time pruža brzu mrežnu stranicu.

Kao što je već spomenuto, Gatsby radi na temelju React-a te kombinira vrhunske tehnologije kao što su GraphQL, Webpack i sl. za stvaranje jedinstvenog iskustva razvojnim programerima.

³⁷ Usp. Gatsby. URL: <https://www.gatsbyjs.com/> (2024-09-01)

³⁸ Usp. Gatsby. How to Source Content from a Headless CMS. URL: <https://www.gatsbyjs.com/docs/how-to/sourcing-data/headless-cms/> (2024-05-19)

Cjelokupni rad Gatsby CMS-a koji obuhvaća dodatke, teme i *strartere* omogućuje izgradnju i implementaciju u nekoliko minuta za ubrzani rad mrežnih mjesta i aplikacija.

Neki od korisnika Gatsby CMS-a su IBM, Jaxxon, Daniel Wellington, moneygeek, clearlink, Little Caesars, bussines.com.³⁹

Izrada mrežnih mjesta

Gatsby pruža mogućnost izrade brze i sigurne mrežne stranice koristeći okvir koji se temelji na Reactu, kao i inovativni podatkovni sloj koji integrira različite sadržaje, API-je i usluge. Taj objedinjeni podatkovni sloj omogućuje kombiniranje podataka iz različitih izvora te ih prikazuje zajedno.

Nadalje, Gatsby kombinira generiranje statičke mrežne lokacije, generiranje statičke stranice i inteligentno prikazivanje stranice za selektivno prethodno učitavanje sadržaja koji je važan što omogućava brzu i učinkovitu mrežnu stranicu.⁴⁰

STRAPI

Strapi je CMS koji se koristi za održavanje i upravljanje Gatsby statičkom mrežnom stranicom. Strapi i Gatsby zajedno čine brzu kombinaciju, Strapi pokreće moderni tehnološki skup koji koristi Node.js, a može i proizvesti sadržaj s GraphQL-om.

Prednosti koje generator statičkih stranica nudi su performanse, sigurnost te niži trošak skaliranja.

Također, dostupna je i ugrađena značajka *webhook* koja omogućava automatsku izgradnju Gatsby projekta nakon ažuriranja podataka; odnosno, nakon napravljenih izmjena u Strapiju, novi sadržaj je odmah aktivan.

³⁹ Usp. Gatsby. URL: <https://www.gatsbyjs.com/> (2024-05-19)

⁴⁰ Usp. Isto.

Prednosti korištenja Strapi CMS-a su brza izgradnja mrežnog mjesta, potpunu optimizaciju aplikacija te sigurnost prema zadanim postavkama.⁴¹

Gatsby kombinira kontrolu i skalabilnost dinamički prikazanih stranica s brzinom generiranja statičkih stranica. Omogućuje stvaranje mrežnih mjesta s visokim učinkom i bogatim sadržajem.

Neke od karakteristika Gatsby-ja su:

- učitavanje podataka iz bilo kojeg izvora podataka,
- mrežne stranice su potpuno funkcionalne React aplikacije što omogućuje stvaranje visokokvalitetnih, dinamičkih mrežnih aplikacija (blogovi, mrežne stranice za e-trgovinu itd.),
- mogućnost odabiranja alternativnih opcija prikazivanja; odgođeno statičko generiranje (DSG) i prikazivanje na strani poslužitelja (SSR) uz statičko generiranje mrežnog mjesta (SSG) omogućuju optimizaciju izvedbe i produktivnosti
- automatizirano dijeljenje koda, optimizacija slike, odgođeno učitavanje i sl.
- Gatsby mrežna mjesta ne zahtijevaju poslužitelje što omogućuje korištenje Gatsby Clouda⁴²

Gatsby obavlja veliki dio posla optimizacije performansi te ima širok raspon dodatka koji pomažu kod upravljanja (SEO). Još neki primjeri generatora statičkih mrežnih stranica su Jekyll i Next JS.⁴³

Gatsby se smatra generatorom statičkih stranica; lako, jednostavno i brzo kreira mrežnu stranicu, a u isto vrijeme podržava tehnologije potrebne korisnicima, radi na svim poslužiteljima, ne zahtijeva unaprijed instalirane baze podataka, nema zaporke, nema upravljačku ploču, pruža

⁴¹ Usp. Strapi. Integrate Gatsby with Strapi. URL: <https://strapi.io/integrations/gatsby-cms> (2024-05-19)

⁴² Usp. Github. Gatsby. URL: <https://github.com/gatsbyjs/gatsby> (2024-05-19)

⁴³ Usp. Hackernoon. How to Build a Gatsby Static Site Using Ghost as a Headless CMS, 2021. URL: <https://hackernoon.com/how-to-build-a-gatsby-static-site-using-ghost-as-a-headless-cms-1k2o35sw> (2024-05-19)

potpunu sigurnost i visoke performanse. Tekst se može pisati i uređivati u *Markdownu* te se pretvarati u HTML kodirane stranice. ⁴⁴

GATSBY I GRAPHQL

Kod kreiranja projekta ili mrežnog mjesta s Gatsbyjem, podacima se pristupa putem upitnog jezika GraphQL-a koji omogućuje deklarativno izražavanje potrebe za podacima. Potreba za podacima se izražava pomoću upita:

```
{
  site {
    siteMetadata {
      title
    }
  }
}
```

Koji vraća podatke:

```
{
  "site": {
    "siteMetadata": {
      "title": "A Gatsby site!"
    }
  }
}
```

⁴⁴ Usp. Medium. Why should I use GatsbyJS instead of WordPress and Ghost CMS for build my website, 2021. URL: <https://abdelhadi.medium.com/why-i-use-gatsby-for-my-site-instead-of-wordpress-and-ghost-cms-520895d6a836> (2024-05-19)

GraphiQL je korisničko sučelje koje vam omogućuje:

- pokretanje upita prema podacima u pregledniku
- istraživanje strukture podataka koji su dostupni preko istraživača vrsta podataka⁴⁵

⁴⁵ Usp. Gatsby. Documentation. GraphQL & Gatsby. URL: <https://www.gatsbyjs.com/docs/graphql/> (2024-09-27)

5. GHOST

Ghost je besplatna platforma za blogove otvorenog koda napisana u JavaScriptu i distribuirana pod licencom MIT-a te je dizajnirana da pojednostavi proces mrežnog objavljivanja za pojedinačne blogere ali i online publikacije.⁴⁶

Ghost je aplikacija za kreatore novih medija za objavljivanje, dijeljenje i razvoj poslovanja oko sadržaja. Sadrži moderne alate za izradu mrežnih stranica i objavljivanje sadržaja.

Ghost omogućuje objavljivanje putem biltena, putem *weba* i e-pošte, dijeljenje sadržaja putem medija i dinamičkih kartica, odnosno dijeljenje sadržaja slikama, GIF-ovima, video zapisima, zvukom, različitim datotekama itd. Objavljivanje putem biltena uključuje dostavljanje sadržaja putem e-pošte kako bi se korisnike obavijestilo o novim objavama i sadržajima.⁴⁷

Na Ghost službenoj stranici dostupan je sadržaj za kreatore (youtubere, voditelje podcasta, glazbenike i umjetnike), za izdavače (autori, novinari, novi mediji), poslovni sadržaj (moderne marke i tvrtke s ambicioznim sadržajnim marketingom) i za razvojne programere (izvorni kodovi, dokumentacija i upute). U dijelu za razvojne programere dostupne su upute za instalaciju na svim operativnim sustavima kao i upute za rad na Ghost projektima. Dostupni tutorijali podijeljeni su u nekoliko razina: najkorisnije (instalacija lokalno, osnovne informacije pri izgradnji Ghost teme, kako otvoriti temu u uređivaču koda, kako dodati ikone i sl.), osnove (preuzimanje uređivača koda, preuzimanje i učitavanje teme, kako poslati vlastitu elektroničku poštu dobrodošlice itd.), nivo iznad (dodavanje HTML i CSS elemenata, vlastita forma za prijavu, vlastite postavke i sl.), napredni nivo (otklanjanje neispravnosti, tablica sadržaja, promjena redoslijeda itd.)⁴⁸

Nadalje, u dokumentaciji za razvojne programere dostupni su izvori za programere (starter početni okvir, API dokumentacija, JAMstack *frontend* okviri); u izvorima za starter početni okvir

⁴⁶ Usp. Aspire Themes. Ghost themes for your blog, newsletter, podcast, portfolio, or news website. Accessible, simple, and easy to customize. URL:

https://aspirethemes.com/?gclid=CjwKCAjwsvujBhAXEiwA_UXnANqRPQTUwtTth9VOSAKV4uQ_d7eOCwYOS2FdidREmSCCPGueezHQeRoCiNwQAvD_BwE (2024-05-19)

⁴⁷ Usp. Ghost. Independent technology for modern publishing. URL: <https://ghost.org/> (2024-05-19)

⁴⁸ Usp. Ghost. Developer docs. URL: <https://ghost.org/docs/> (2024-05-19)

dostupna je dokumentacija i šablona na githubu koju je moguće preuzeti, u API dokumentaciji dostupni su primjeri i objašnjenja korištenja istih, a u JAMstack su upute i objašnjenja za korištenje *frontend* okvira.⁴⁹

Kao što je već navedeno, Ghost je besplatna platforma za blogove otvorenog koda, a stvorena je kao jednostavnija alternativa WordPressu. Osnivač Ghosta je John O'Nolan a njegov je cilj stvoriti pojednostavljeno iskustvo usredotočeno na pisanje. Tako je osmišljen Ghost, novi CMS sustav izgrađen na modernijim tehnologijama te napisan u Node.js. Ghost također ima širok raspon API-ja koji su jednostavni za korištenje.⁵⁰

Ghost CMS je nastao na temelju WordPressa, odnosno bivši razvojni programer Wordpressa izgradio je sličnu, ali jednostavniju i prilagođenu verziju kako bi to bila platforma za pisanje blogova, a ne kao sustav za upravljanje sadržajem.

Značajke Ghosta:

- besplatno je, otvorenog koda
- brz sustav, izgrađen na modernim tehnologijama koje omogućuju brže vrijeme učitavanja
- jednostavno, sigurno i fleksibilno
- vrlo dobro korisničko iskustvo, uređivač teksta podržava HTML, slike i sl.
- mogućnost pisanja i uređivanja od strane više osoba
- brzo i optimizirano arhiviranje (SEO)
- omogućuje API-je za mrežno mjesto koje se može povezati s bilo kojim drugim softverom
- ne podržava RTL s desna na lijevo
- velika količina zauzeća diska – oko 1 GB
- upravljačka ploča ne sadrži arapski jezik⁵¹

⁴⁹ Usp. Ghost. Ghost on the JAMstack. URL: <https://ghost.org/docs/jamstack/> (2024-07-15)

⁵⁰ Usp. Hackernoon. How to Build a Gatsby Static Site Using Ghost as a Headless CMS, 2021. URL: <https://hackernoon.com/how-to-build-a-gatsby-static-site-using-ghost-as-a-headless-cms-1k2o35sw> (2024-05-19)

⁵¹ Usp. Medium. Why should I use GatsbyJS instead of WordPress and Ghost CMS for build my website, 2021. URL: <https://abdelhadi.medium.com/why-i-use-gatsby-for-my-site-instead-of-wordpress-and-ghost-cms-520895d6a836> (2024-05-19)

GHOST ADMINISTRATIVNO SUČELJE

Ghost administrativno sučelje ne podržava prijevod na druge jezike, odnosno administrativno sučelje dostupno je isključivo na engleskom jeziku. No, u ožujku 2023. godine omogućeno je prevođenje ključnih komponenti, uključujući portal za članove. Beta verzija te značajke pokrenuta je lipnju 2023. godine i dostupna je svim korisnicima Ghost CMS-a. Podržano je preko 40 jezika, a uključuje prijevod za portal za članove, pretraživanje i komentare (slika 5.).⁵² Među podržanim jezicima nalazi se i hrvatski jezik.⁵³



The image shows a registration form for a members portal titled "Sportfest". The form includes a close button (X) in the top right corner. It features two input fields: "Ime i prezime" (Name and surname) with the placeholder text "Jamie Larson", and "E-pošta" (Email) with the placeholder text "jamie@example.com". Below the input fields is a prominent pink button labeled "Registracija" (Registration). At the bottom of the form, there is a link that says "Već imate račun? [Prijava](#)" (Already have an account? [Login](#)).

Slika 5. portal za članove

⁵² Usp. GhostCave. How to Translate Ghost CMS Members Portal into Your Language. URL: <https://ghostcave.org/how-to-translate-ghost-cms-members-portal-into-your-language/> (2024-09-27)

⁵³ Usp. Try Ghost. Ghost. URL: <https://github.com/TryGhost/Ghost/tree/main/ghost/i18n/locales?ref=ghostcave.org> (2024-09-27)

Pokretanje značajke odvija se u postavkama Ghost administrativnog sučelja gdje se otvara *Labs Menu* i uključuje se *Portal translation*. Nakon toga u postavkama pod opcijom *General* mijenja se *Publication language* na jezik po izboru. Ghost Admin još ne podržava internacionalizaciju. Što se tiče višezječnog sadržaja, podržano je, no ako se planira objavljivati na više jezika, preporuka je da se koristi jedna instalacija Ghost-a po jeziku, što je lako za pokretanje ali i dugoročno održivo.⁵⁴

⁵⁴ Usp. Ghost. Translations in Ghost. URL: <https://ghost.org/docs/faq/translation/> (2024-09-27)

6. GATSBY + GHOST

Korištenje statičkog generatora mrežnog mjesta uključuje lokalno pohranjivanje *markdown* datoteka u repozitorij koda. To uključuje korištenje uređivača koda za pisanje sadržaja i GIT tijekom rada za objavljivanje što dobro funkcionira za male mrežne stranice ili razvojne programere. No, nije idealan za profesionalne izdavače gdje se koristi bezglavi CMS.

Tako se Ghost može koristiti za autorstvo, nakon toga slijedi izgradnja *frontend* dijela u Gatsbyju gdje se povlači sadržaj iz Ghost API-ja. Neke od prednosti toga za izdavače su:

- razvojni programeri koriste svoj željeni skup
- autori imaju preferiranog urednika i upravitelja sadržajem
- performanse su na maksimumu
- sigurnost je na maksimumu
- stranica je skalabilna

Izrada mrežnih stranica na takav način poznata je kao JAMstack (JavaScript, API, Markup).⁵⁵

IZDAVAŠTVO USMJERENO NA BUDUĆNOST

Objedinjavanje različitih tehnologija u jednom sučelju centralizira inače decentralizirane usluge na način koji prvenstveno zadovoljava potrebe vlasnika stranice te potiče fleksibilnost i skalabilnost.

Ne postoje dodatci i proširenja specifičnih za platformu; samo je jedan *frontend* i mnogo API-ja, a sve zajedno služi kao jedno mrežno mjesto ili aplikacija s Gatsbyem.⁵⁶

⁵⁵ Usp. Gatsby. Ghost. URL: <https://gatsby.ghost.org/about/> (2024-08-05)

⁵⁶ Usp. Isto.

INSTALACIJA GHOST-a LOKALNO

Potrebno je preuzeti Node.js i node package manager, podržana verzija 18.x, a preporučena verzija je 20.x. Nakon toga, preko naredbenog retka instalira se Ghost CMS pomoću naredbe

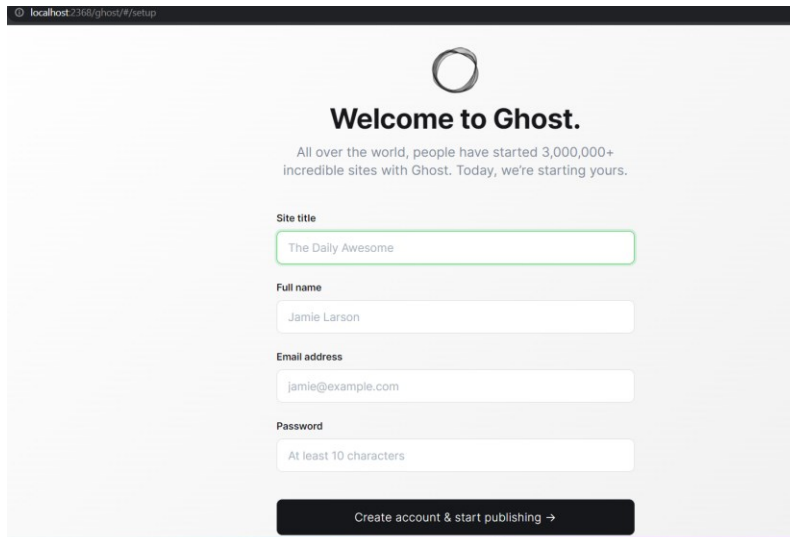
```
npm install ghost-cli -g
```

nakon toga, pozicioniramo su u prazan direktorij te instaliramo naredbom

```
ghost install local
```

nakon instalacije, dobivamo link <http://localhost:2368> koji nas vodi na mrežnu stranicu.⁵⁷

Kad se kreira direktorij, otvaramo Ghost na <http://localhost:2368/ghost/> ; pojavljuje se početna stranica za kreiranje mrežnog mjesta, potrebno je kreirati korisnički račun kako bi se otvorilo administrativno sučelje za upravljanje mrežnim mjestom (slika 6.).



The screenshot shows the Ghost local installation setup page. At the top, there is a Ghost logo and the text "Welcome to Ghost." Below this, a message states: "All over the world, people have started 3,000,000+ incredible sites with Ghost. Today, we're starting yours." The form contains the following fields:

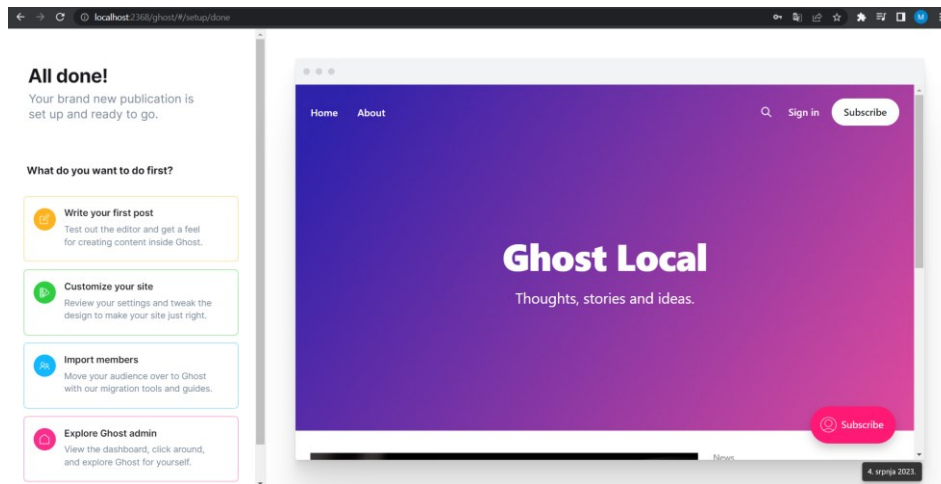
- Site title:** The Daily Awesome
- Full name:** Jamie Larson
- Email address:** jamie@example.com
- Password:** At least 10 characters

At the bottom of the form is a button labeled "Create account & start publishing →".

Slika 6. kreiranje profila

⁵⁷ Usp. Ghost. How to install Ghost locally. URL: <https://ghost.org/docs/install/local/> (2024-08-20)

Nakon kreiranja korisničkog računa, otvara se stranica s predlošcima pomoću koje se kreira vlastiti sadržaj (slika 7.).



Slika 7. Početna stranica mrežnog mjesta

Nadalje, za *frontend* dio mrežne stranice uključujemo Gatsby.js. Za početak, potrebno je generirati novi projekt pomoću predloška Gatsby Starter Ghost sa naredbom:

```
gatsby new my-gatsby-site https://github.com/TryGhost/gatsby-starter-ghost.git
```

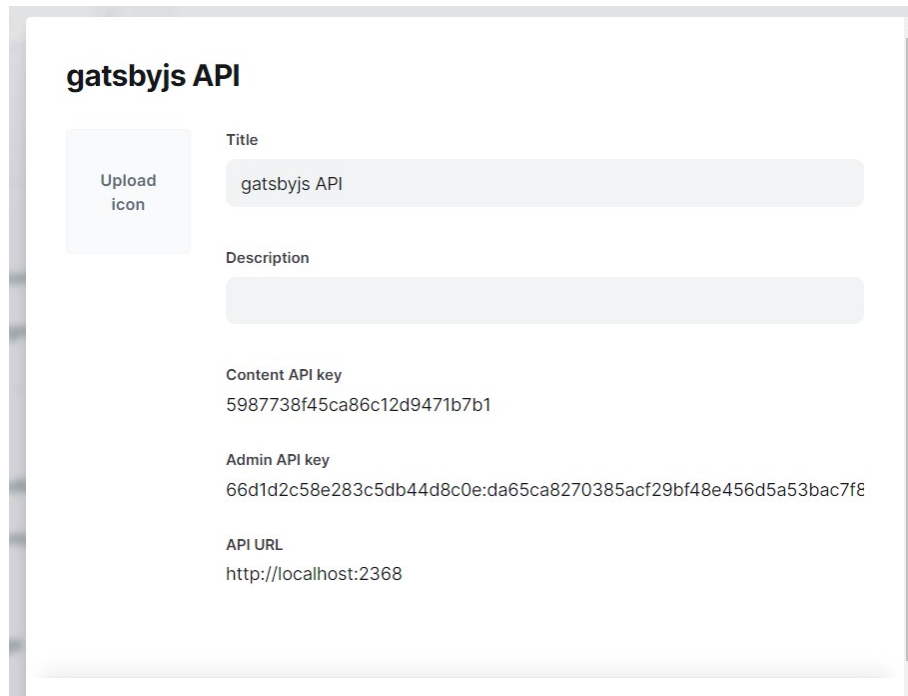
Prije razvoja i prilagodbe nove stranice, potrebno je provjeriti je li sve uspješno instalirano pomoću naredbe:

```
gatsby develop
```

Nakon toga, ako je sve uspješno instalirano, na linku <http://localhost:8000/> se otvara nova Gatsby mrežna stranica.⁵⁸

⁵⁸ Usp. Ghost. Working with Gatsby. URL: <https://ghost.org/docs/jamstack/gatsby/> (2024-08-20)

Nadalje, kako bi se implementirao Gatsbyjs API u Ghost CMS u izborniku preko opcije *integrations* dodaje se novi predložak gdje se kreiraju podaci (slika 8.)



The screenshot shows a form titled "gatsbyjs API" with the following fields and values:

- Title:** gatsbyjs API
- Description:** (empty text area)
- Content API key:** 5987738f45ca86c12d9471b7b1
- Admin API key:** 66d1d2c58e283c5db44d8c0e:da65ca8270385acf29bf48e456d5a53bac7f8
- API URL:** http://localhost:2368

There is also an "Upload icon" button on the left side of the form.

Slika 8. generiranje API ključa

Koji se onda dodaju u izvorni kod (slika 9.)

```
{  
  "development": {  
    "apiUrl": "http://localhost:2368",  
    "contentApiKey": "5987738f45ca86c12d9471b7b1"  
  },  
}
```

Slika 9. Povezivanje Ghost i Gatsby putem API ključa

Nadalje, uređivanje i dodavanje sadržaja implementira se preko Ghost-a, gdje se sadržaj uredi te se preko naredbenog retka pokreće i ažurira na Gatsby mrežnom mjestu (slika 10.)



Slika 10. Uređivanje mrežnog mjesta

Postoje brojne opcije uređivanja, dodavanja i prilagođavanja sadržaja kao što je uređivanje objava, dodavanje fotografija, izrada i prilagođavanje vlastitog dizajna, mijenjanje fonta i boje, dodavanje vlastitog loga itd.

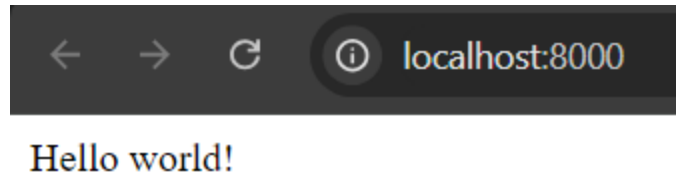
GATSBY MREŽNO MJESTO

Nadalje, kao primjer razvoja Gatsby mrežnog mjesta u ovom radu, korišten je gatsby starter hello world predložak. Pomoću naredbe

gatsby new gatsby-starter-hello-world <https://github.com/gatsbyjs/gatsby-starter-hello-world>⁵⁹

Dalje, naredbom gatsby develop kreira se mrežno mjesto na linku <http://localhost:8000/> (slika 10.)

⁵⁹ Usp. Gatsby. Gatsby Starter Library. URL: <https://www.gatsbyjs.com/starters/gatsbyjs/gatsby-starter-hello-world> (2024-08-22)



Slika 11. Hello World predložak mrežnog mjesta

Nakon instalacije, izvorni kod uređujemo u uređivaču koda, u ovom slučaju korišten je Visual Studio Code.

Zatim u datoteci index.js uređujemo kod pomoću HTML i CSS elemenata (slika 11.)

```
JS index.js ×
src > pages > JS index.js > ...
1  import * as React from "react"
2
3  export default function Home() {
4    return <div style={{color: 'lightgray', backgroundColor: 'darkgray'}}>
5      <h1>IMPLEMENTACIJA GATSBY I GHOST CMS TEHNOLOGIJA PRI IZRADI MREŽNOG MJESTA</h1>
6      <p>diplomski rad</p>
7    </div>
8  }
9
```

Slika 12. Dodavanje HTML i CSS elemenata

Stranicu možemo izgraditi pomoću naredbe `npm run build` u naredbenom retku, te nakon svake promjene u izvornom kodu pomoću iste naredbe nadogradimo mrežnu stranicu.

KOMPONENTE

Komponente su temeljni dijelovi arhitekture temeljeni na React-u, dijelovi su korisničkog sučelja za višekratnu upotrebu a pomažu pri organizaciji i strukturi mrežnog mjesta. (slika 13.)

```
src > novidirektorij > JS brojac.js > default
1  import * as React from "react"
2
3  class Counter extends React.Component {
4    constructor() {
5      super()
6      this.state = { count: 0 }
7    }
8    render() {
9      return <div>
10         <h1>Brojač</h1>
11         <p>current count: {this.state.count}</p>
12         <button onClick={() => this.setState({count: this.state.count -1})
13         }>minus</button>
14         <button onClick={() => this.setState({count: this.state.count +1})
15         }>plus</button>
16         </div>
17       }
18     }
19   }
20 }
21
22 export default Counter
```

Slika 13. Primjer komponente

Ako navedenu komponentu želimo dodati na početnu stranicu, odnosno početnu datoteku index.js to činim putem naredbe (slika 14.)

```
import * as React from "react"
import { Link } from "gatsby"
import { Brojac } from "./brojac";
```

Slika 14. implementacija komponente

Nekoliko je vrsta komponenti koje Gatsby koristi, a to su:

- funkcionalne komponente – jednostavne JavaScript funkcije koje vraćaju JSX (JavaScript XML)
- komponente rasporeda – upravljanje strukturom stranice – podnožje, zaglavlje itd.
- komponente stranice – svaka datoteka u src/ime-direktorija postaje ruta
- dijeljene komponente – gumb, obrazac i sl.⁶⁰

GraphQL u komponentama – komponente dohvaćaju podatke putem GraphQL upita što omogućuje dohvaćanje sadržaja iz markdown datoteka, CMS sustava ili drugih izvora.⁶¹

Komponente rasporeda

Gatsby prema zadanim postavkama ne primjenjuje automatski raspored na stranice, ali slijedi Reactov kompozicijski model dodavanja i korištenja komponenti što omogućuje stvaranje rasporeda stranice na više razina; zaglavlje i podnožje, izbornik itd.

Komponente rasporeda su namijenjene odjeljcima mrežnog mjesta koje želimo da se pojavljuju na više stranica. Na primjer, Gatsby mrežne stranice će imati većinom zajedničko zaglavlje i podnožje (slika 15).⁶²

```
import React from "react"

export default function Layout({ children }) {
  return (
    <div style={{ margin: `0 auto`, maxWidth: 650, padding: `0 1rem` }}>
      {children}
      <h2>Podnožje</h2>
    </div>
  )
}
```

Slika 15. Komponenta rasporeda

⁶⁰ Usp. Gatsby. Documentation. Building with Components. URL:

<https://www.gatsbyjs.com/docs/conceptual/building-with-components/> (2024-09-27)

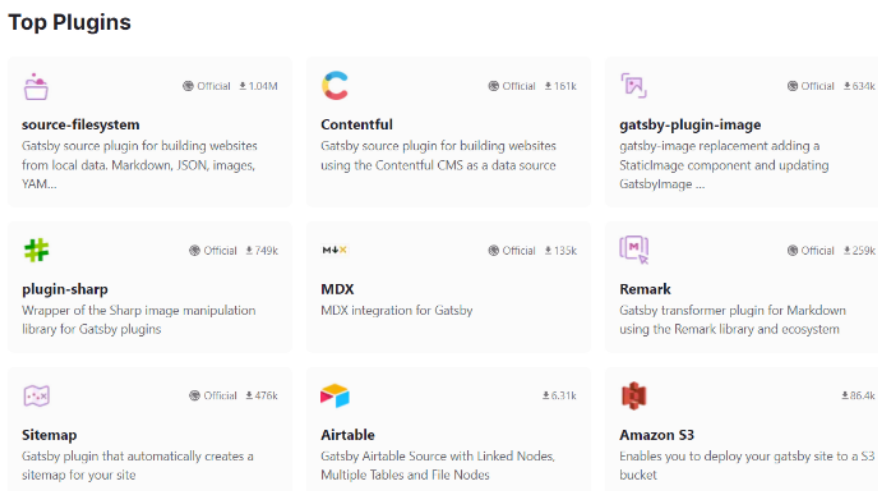
⁶¹ Usp. Gatsby. Documentation. GraphQL & Gatsby. URL: <https://www.gatsbyjs.com/docs/graphql/> (2024-09-27)

⁶² Usp. Gatsby. Documentation. Layout Components. URL: <https://www.gatsbyjs.com/docs/how-to/routing/layout-components/> (2024-08-22)

DODATCI

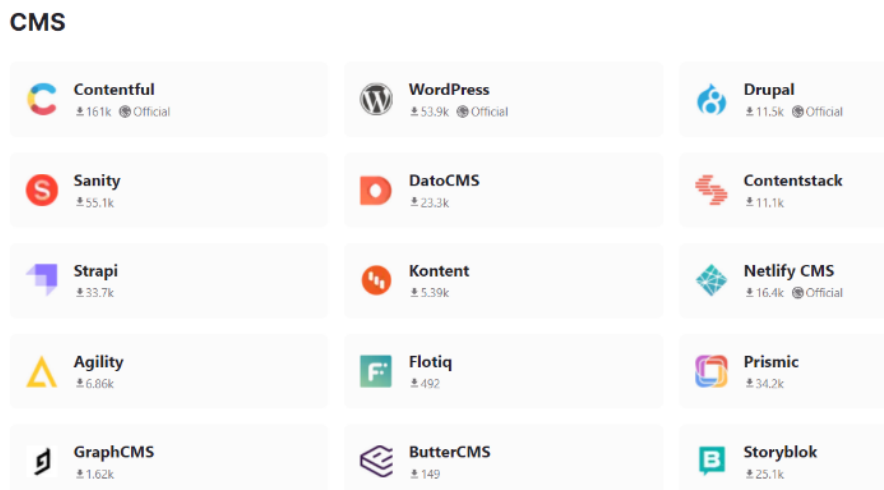
Nadalje, možemo uključiti različite dodatke koji se nalaze na Gatsby službenoj stranici, a uključuju:

Top Plugins (slika 16.)



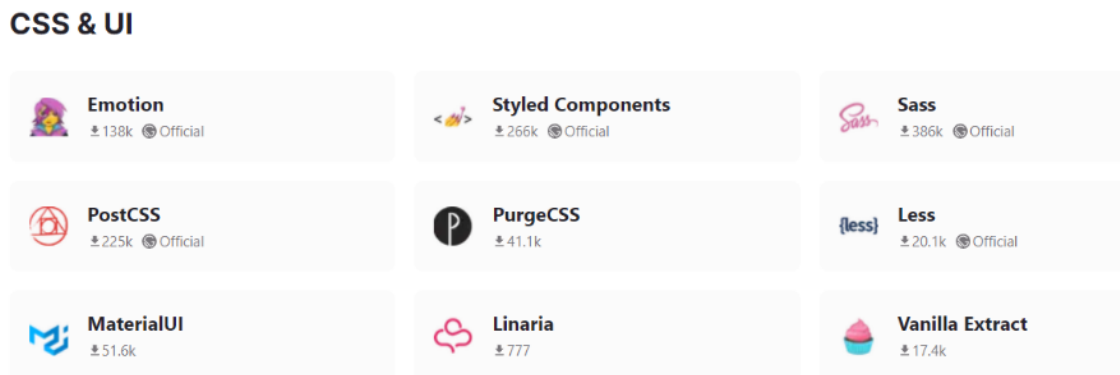
Slika 16. prikazuje najčešće korištene dodatke

CMS dodaci (slika 17.)



Slika 17. prikazuje CMS dodatke

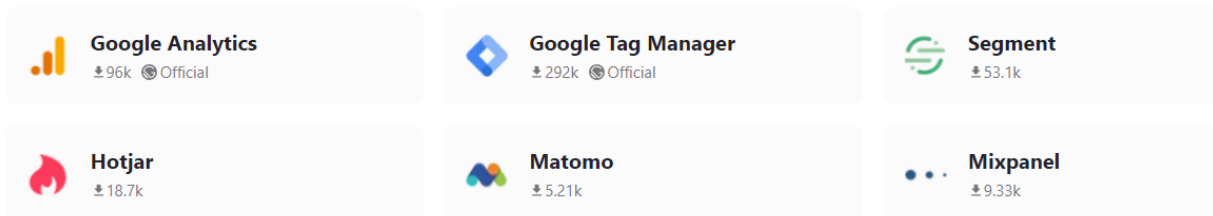
CSS & UI dodaci(slika 18.)



Slika 18. prikazuje CSS i UI dodatke

Analytics (slika 19.)

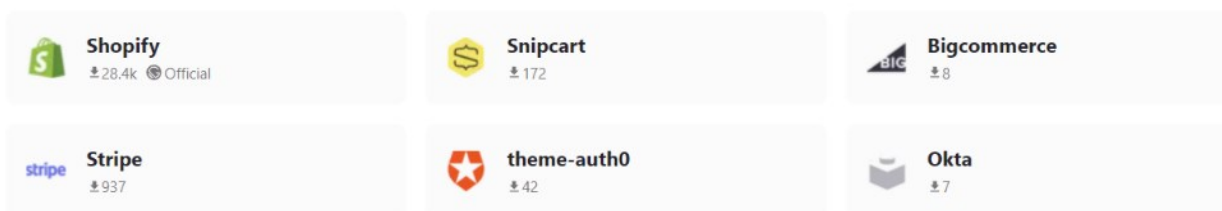
Analytics



Slika 19. prikazuje analitičke dodatke

E-commerce, Payment & Auth (slika 20.)

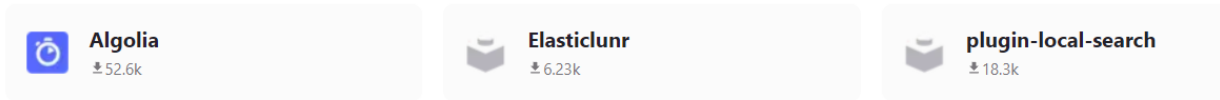
E-commerce, Payment & Auth



Slika 20. prikazuje dodatke koji se odnose na e-trgovinu, plaćanje u autorizaciju

Search (slika 21.)

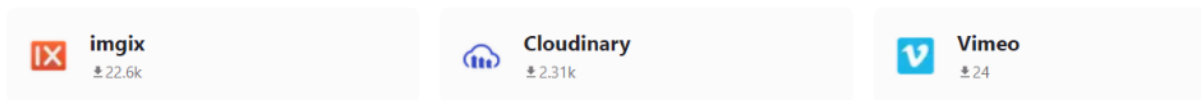
Search



Slika 21. prikazuje dodatke koji se odnose na pretraživanje

Media (slika 22.)

Media



Slika 22. prikazuje dodatke koji se odnose na različite medije

Također, postoji mogućnost kreiranja i vlastitog dodatka.⁶³

U radu je kao primjer korišten CSS & UI dodatak *Styled Components*, a on se instalira naredbom `npm install gatsby-plugin-styled-components styled-components babel-plugin-styled-components` uređuje se u datoteci `gatsby-config.js` (slika 23.)⁶⁴

⁶³ Usp. Gatsby. Welcome to the Gatsby Plugin Library. URL: <https://www.gatsbyjs.com/plugins> (2024-08-20)

⁶⁴ Usp. Gatsby. Gatsby-plugin-styled-components. URL: <https://www.gatsbyjs.com/plugins/gatsby-plugin-styled-components> (2024-08-20)

```

module.exports = {
  plugins: [
    {
      resolve: `gatsby-plugin-styled-components`,
      options: {}
    }
  ],
}

```

Slika 23. Implementacija dodatka

METAPODACI

Za dijelove mrežnog mjesta, kao što su naslov, autor itd. koje želimo ponovno upotrijebiti, pohranjujemo ih kao metapodatke u gatsby-config.js datoteci. Na taj način podatke spremamo na jedno mjesto i ako je potrebno izmijeniti, dovoljno je samo na jednom mjestu ažurirati podatke (slika 24.)⁶⁵

```

module.exports = {
  siteMetadata: {
    title: "IMPLEMENTACIJA GATSBY I GHOST CMS TEHNOLOGIJA PRI IZRADI MREŽNOG MJESTA",
    author: "Monika Pavić"
  },
  plugins: [
    {
      resolve: `gatsby-plugin-styled-components`,
      options: {}
    }
  ],
}

```

Slika 24. Dodavanje metapodataka

⁶⁵ Usp. Gatsby. Documentation. URL: <https://www.gatsbyjs.com/docs/reference/config-files/gatsby-config/> (2024-08-22)

Kako bi se tim podacima pristupilo, koristi se GraphQL upitni jezik. Gatsby organizira sve podatke i omogućava njihovo otkrivanje pomoću alata GraphiQL. GraphiQL je korisničko sučelje koje omogućuje pokretanje upita prema podacima u pregledniku, istraživanje strukture podataka koji su dostupni putem istraživača vrsta podataka a može mu se pristupiti putem linka localhost:8000/___graphql

U izvornom kodu ga dodajemo na način (slika 25. i 26.):

```
export const query = graphql`
- # query will go here
+ query HomePageQuery {
+   site {
+     siteMetadata {
+       title
+       author
+     }
+   }
+ }
```

Slika 25. Implementacija GraphQL

```
export default function Home({data}) {
  return <div style={{color: 'lightgray', backgroundColor: 'darkgray'}}>
    <h1>{data.site.siteMetadata.title}</h1>
    <br />
    <p>{data.site.siteMetadata.author}</p>
    <br />
    <Link to="/cms-tehnologija/">CMS tehnologija</Link>
    <br />
    <Brojac color="white"/>
  </div>
}
```

Slika 26. Implementacija GraphQL

7. APLIKACIJA

U ovom poglavlju prikazan je konkretan primjer aplikacije koja spaja sve što je prethodno objašnjeno. Aplikacija omogućuje CRUD (*create, read, update i delete*) članka, a dostupna je na linku <https://github.com/mpavic2/diplomski.git>. Tema aplikacije su sportski događaji, odnosno aplikacija omogućuje otkrivanje i praćenje sportskih događanja.

Prije nego što se opiše samo kreiranje aplikacije, potrebno je objasniti uvjete za uspješno kreiranje iste. Ono što je bila najveća problematika ove aplikacije je verzija node.js – a. na službenoj stranici Ghost CMS-a navedene su podržane verzije; iako je verzija 18.x podržana za uspješno pokretanje Ghost-a ⁶⁶i prilikom istog nema nikakvih grešaka, samo upozorenja, problem se javio prilikom povezivanja Gatsby i Ghost CMS sustava. Rješenje problema koji se pojavio je ažuriranje node.js-a na verziju 20.x, koja je ujedno i preporučena verzija za korištenje.

Za početak je potrebno, kao i u prethodnim primjerima, pripremiti Ghost i Gatsby okruženje. Prvo se instalira Ghost lokalno i pokrene se; potrebno je imati instalirano:

- node.js – 20.x verzija
- npm
- Ghost CLI - `npm install -g ghost-cli`

Nakon instalacije Ghost CLI u naredbenom retku, kreira se direktorij gdje se instalira Ghost, nakon pozicioniranja u isti, pokrećemo naredbu

```
ghost install
```

koja preuzima i instalira Ghost CMS i potrebne ovisnosti.

Naredbom `ghost start` pokreće se Ghost na <http://localhost:2368/> gdje se kreira profil (slika 6.) te se pojavi administrativno sučelje (slika 7.).

Nakon toga je potrebno kreirati *API key* (slika 8.) Nakon toga slijedi instalacija Gatsby-ja; potrebno je imati instalirano:

- node.js – 20.x verzija

⁶⁶ Usp. Ghost. Developer docs. Supported Node versions. URL: <https://ghost.org/docs/faq/node-versions/#> (2024-09-24)

- npm
- Gatsby CLI - npm install -g gatsby-cli

slijedi kreiranje novog projekta kao i dodatka koji uključuje Ghost, a koristio se gatsby-starter-ghost sa službene Gatsby stranice koji je posljednji put ažuriran u srpnju ove godine. Instalacija istog ide preko naredbe

```
gatsby new gatsby-starter-ghost https://github.com/TryGhost/gatsby-starter-ghost
```

nakon toga potrebno je preuzeti i njegove ovisnosti naredbom

```
npm install
```

nakon toga, pokreće se projekt naredbom

```
gatsby develop
```

i otvara se na <http://localhost:8000/> .

gatsby-starter-ghost uključuje:

- Ghost integracija sa spremnim sadržajem rezerviranog mjesta i podrškom za *webhooks*
- Minimalistički dizajn prilagođen različitim širinama zaslona
- Označavanje stranica za postove, oznake i autore
- Meta prilagođena SEO-u
- JSON-LD shema
- OpenGraph strukturirani podaci
- Meta Twitter kartica
- Generiranje karte mrežnih stranica (*Sitemap*)
- XML Sitemaps
- Progresivna mrežna aplikacija
- Izvanmrežna podrška
- RSS feed
- Netlify integracija spremna za implementaciju

I 25 ovisnosti:

- @tryghost/helpers-gatsby
- gatsby
- gatsby-awesome-pagination
- gatsby-plugin-advanced-sitemap
- gatsby-plugin-catch-links
- gatsby-plugin-feed
- gatsby-plugin-image
- gatsby-plugin-manifest
- gatsby-plugin-offline
- gatsby-plugin-react-helmet
- gatsby-plugin-sharp
- gatsby-source-fileSystem
- gatsby-source-ghost
- gatsby-transformer-sharp
- @tryghost/helpers
- cheerio
- lodash
- react
- react-dom
- react-helmet
- url
- babel-eslint
- eslint
- eslint-plugin-ghost
- eslint-plugin-react⁶⁷

⁶⁷ Usp. Gatsby. Gatsby-starter-ghost. URL: <https://www.gatsbyjs.com/starters/TryGhost/gatsby-starter-ghost> (2024-09-24)

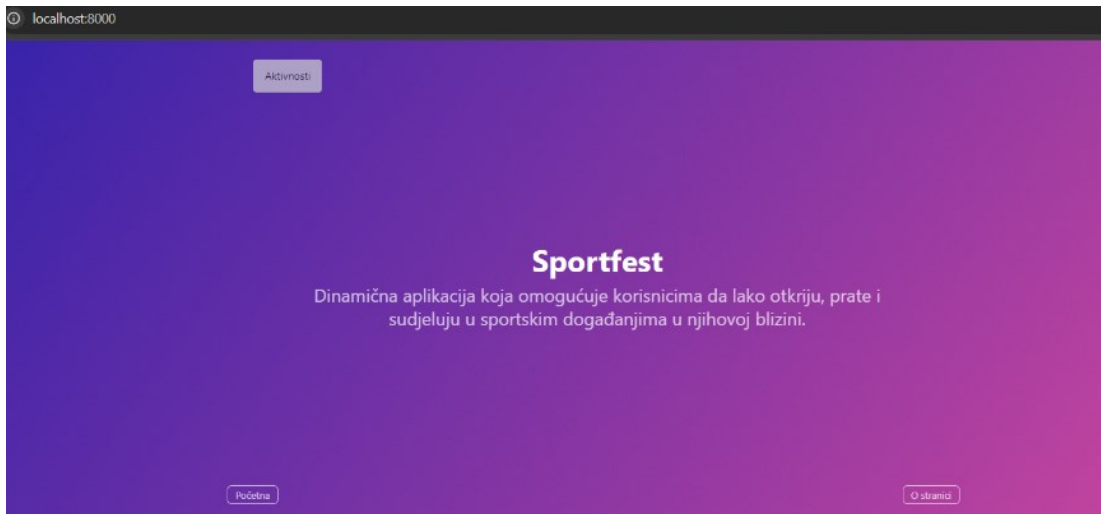
POVEZIVANJE GATBSY I GHOST SUSTAVA

Kako bi se te dvije tehnologije povezale, potrebno je u uređivaču koda, u ovom slučaju *Visual Studio Code-u*, u datoteci `.ghost.json` dodati url na Ghost administrativno sučelje i API ključ (slika 27.) koji se kreira u postavkama Ghost sučelja (slika 7.).

```
diplomski > {} .ghost.json > ...
1  {
2    "development": {
3      "apiUrl": "http://localhost:2368",
4      "contentApiKey": "4b2085237d69778726465c3f61"
5    },
6    "production": {
7      "apiUrl": "http://localhost:2368",
8      "contentApiKey": "4b2085237d69778726465c3f61"
9    }
10 }
```

Slika 27. <https://github.com/mpavic2/diplomski/blob/master/.ghost.json>

Nakon toga, naredbom `gatsby develop` ponovno se pokreće projekt koji povezuje Gatsby *frontend* dio i Ghost *backend* dio, odnosno ono što je dodano preko Ghost administrativnog sučelja sad je vidljivo na <http://localhost:8000/> (slika 28.)



Biciklistička utrka

* Naziv: Utrka oko jezera Jarun * Datum: 28. kolovoza 2024. * Vrijeme: 9:00 - 12:00 * Mjesto: Jezero Jarun, Zagreb * Opis: Sudjelujte u biciklističkoj utrci koja kruži oko prekrasnog jezera Jarun. Najbolje svih dobnih skupina su dobrodošli. Prijavite se za sudjelovanje!

 Monika Pavic

Slika 28. sadržaj iz Ghosta na Gatsby localhostu

KREIRANJE APLIKACIJE

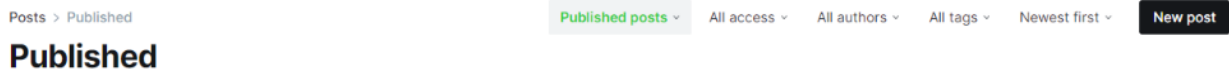
Sljedeće što je potrebno napraviti je kreiranje funkcija za ažuriranje i brisanje objava. Ono što još možemo prikazati je mijenjanje sadržaja unutar koda, odnosno personalizacija mrežnog mjesta. (slika 29.)



Slika 29. <https://github.com/mpavic2/diplomski/blob/master/src/components/common/Layout.js>

Ono što je dalje potrebno objasniti jesu funkcije CRUD aplikacije;

CREATE – kreiranje sadržaja – kreiranje se odrađuje preko Ghost CMS administrativnog sučelja ili unutar Gatsby-ja preko Ghost admin ključa koji se dodaje unutar obrasca za kreiranje nove objave. U prikazanoj aplikaciji nove objave dodane su preko Ghost CMS-a na način da se dodaje nova objava koristeći opciju *New post* (slika 30.)



Slika 30.

Objavu je potrebno kreirati te objaviti kako bi ona bila vidljiva na aplikaciji.

Što se tiče Ghost sustava, kao što je u poglavlju Ghost navedeno, nije moguć prijevod u samom administrativnom sučelju već je omogućen prijevod samo određenih značajki. Ono što je moguće prevesti su značajke i teme na *frontend* dijelu stranice kojom se upravlja na način da se u uređivaču koda otvori Ghost projekt gdje se kreira datoteka koja sadrži prijevod na hrvatski; `content/themes/tema-koja-se-koristi/hr-prijevod.json` u koju se dodaje prevedeni tekst u JSON formatu. Da bi se prijevod upotrijebio u temi, sav statički tekst se prevodi s `{{t}}` *helperom*.⁶⁸ Kako bi se prijevod postavio, potrebno je otići u postavke u Ghost administrativnom sučelju pod opciju *General* i pod sekcijom *Publication Language* postaviti hrvatski jezik (hr) i te promjene spremite nakon čega korištena tema poprima hrvatski prijevod (slika 31.)

⁶⁸ Usp. Ghost. Developer docs. Translate. URL: <https://ghost.org/docs/themes/helpers/translate/> (2024-09-27)

Dinamična aplikacija koja omogućuje korisnicima da lako otkriju, prate i sudjeluju u sportskim događanjima u njihovoj blizini.

monika@primjer.com

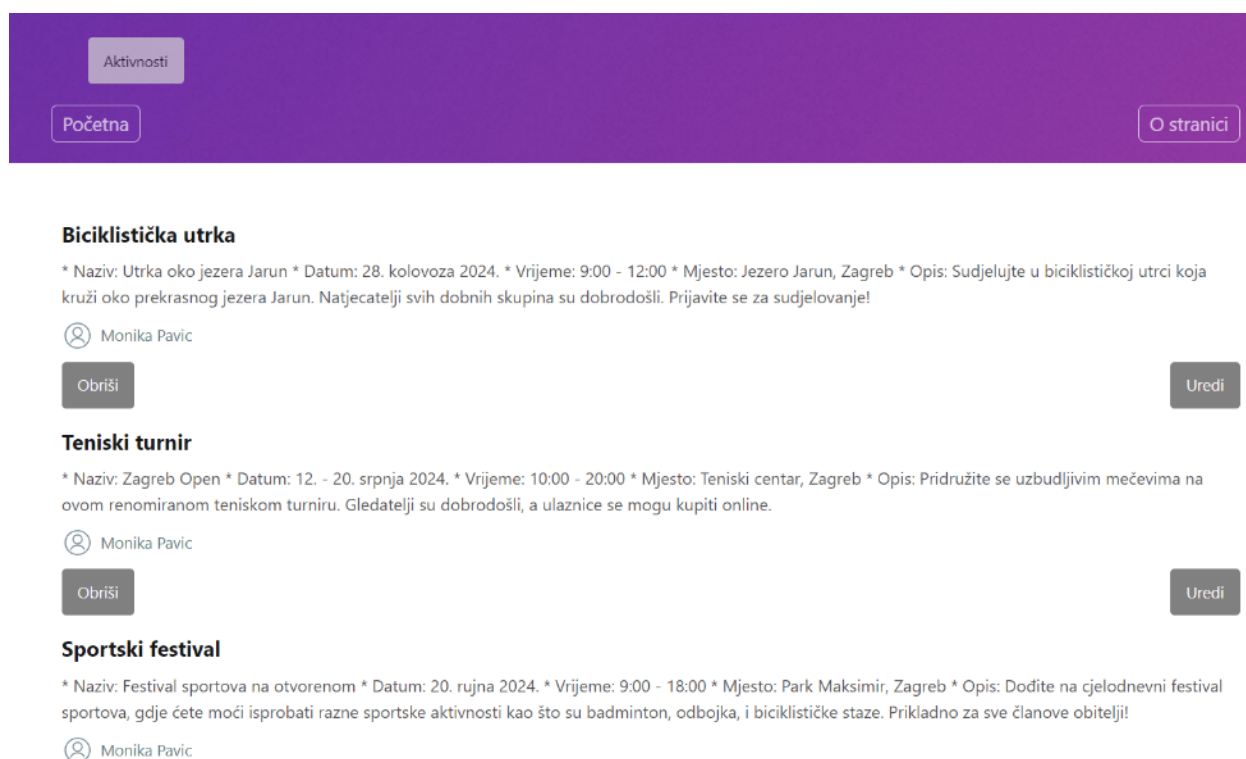
Pretplati se

Slika 31. prijevod Ghost teme na hrvatski jezik

READ – prikaz objava – za prikaz objava dodajemo gumb *Aktivnosti* (slika 32.) i klikom na taj gumb prikazuju se sve objavljene objave (slika 33.). Podaci se prikupljaju putem GraphQL-a u funkciji `pageQuery`; upit traži sve kreirane objave po datumu poredane u DESC redoslijedu, odnosno da su najnovije objave prikazane prve. Prikupljene objave pohranjuju se u varijabli `posts`, a svaka je objava dio niza `edges`. Funkcija `map` prolazi svakom objavom u nizu i za svaku objavu se koristi komponenta `PostCard` gdje se svaka objava prosljeđuje kao `post` atribut. Komponenta `PostCard` zatim prikazuje naslov i sadržaj za svaku objavu.



Slika 32. <https://github.com/mpavic2/diplomski/blob/master/src/pages/posts.js>



Slika 33. prikaz aktivnosti

UPDATE – ažuriranje objava – kreiran je gumb *Uredi* (slika 34.) na svakoj objavi čija je funkcija da omogući ažuriranje sadržaja objave, odnosno pojavi se implementirani WYSIWYG uređivač u kojem mijenjamo naslov i sadržaj (slika 35.). Ažuriranje, odnosno uređivanje naslova i sadržaja objave odvija se na način da funkcija `handleUpdatePost` šalje ažurirane podatke Ghost CMS-u putem HTTP PUT zahtjeva; u API URL-u koristi se ID objave koja se uređuje, a autorizacija se osigurava preko API admin ključa generiranog u Ghost CMS-u.

Biciklistička utrka

* Naziv: Utrka oko jezera Jarun * Datum: 28. kolovoza 2024. * Vrijeme: 9:00 - 12:00 * Mjesto: Jezero Jarun, Zagreb * Opis: Sudjelujte u biciklističkoj utrci koja kruži oko prekrasnog jezera Jarun. Natjecatelji svih dobnih skupina su dobrodošli. Prijavite se za sudjelovanje!

 Monika Pavic

Obrisi

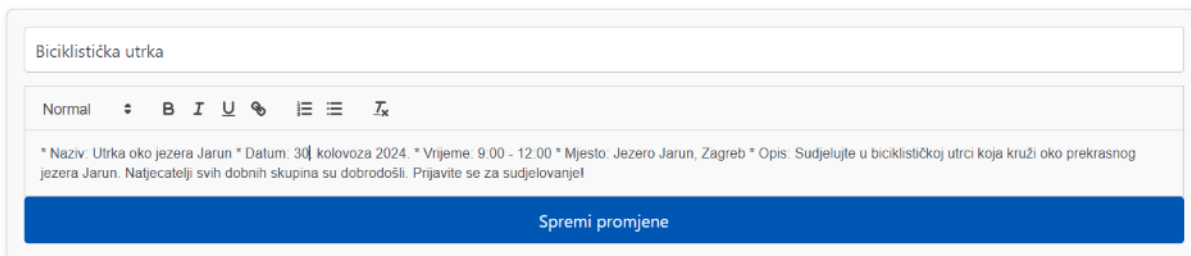
Uredi

Slika 34. <https://github.com/mpavic2/diplomski/blob/master/src/templates/index.js>

Biciklistička utrka

* Naziv: Utrka oko jezera Jarun * Datum: 28. kolovoza 2024. * Vrijeme: 9:00 - 12:00 * Mjesto: Jezero Jarun, Zagreb * Opis: Sudjelujte u biciklističkoj utrci koja kruži oko prekrasnog jezera Jarun. Natjecatelji svih dobnih skupina su dobrodošli. Prijavite se za sudjelovanje!

 Monika Pavic



Slika 35. WYSIWYG uređivač

Postoji nekoliko React WYSIWYG uređivača a u aplikaciji je korišten React Quill;

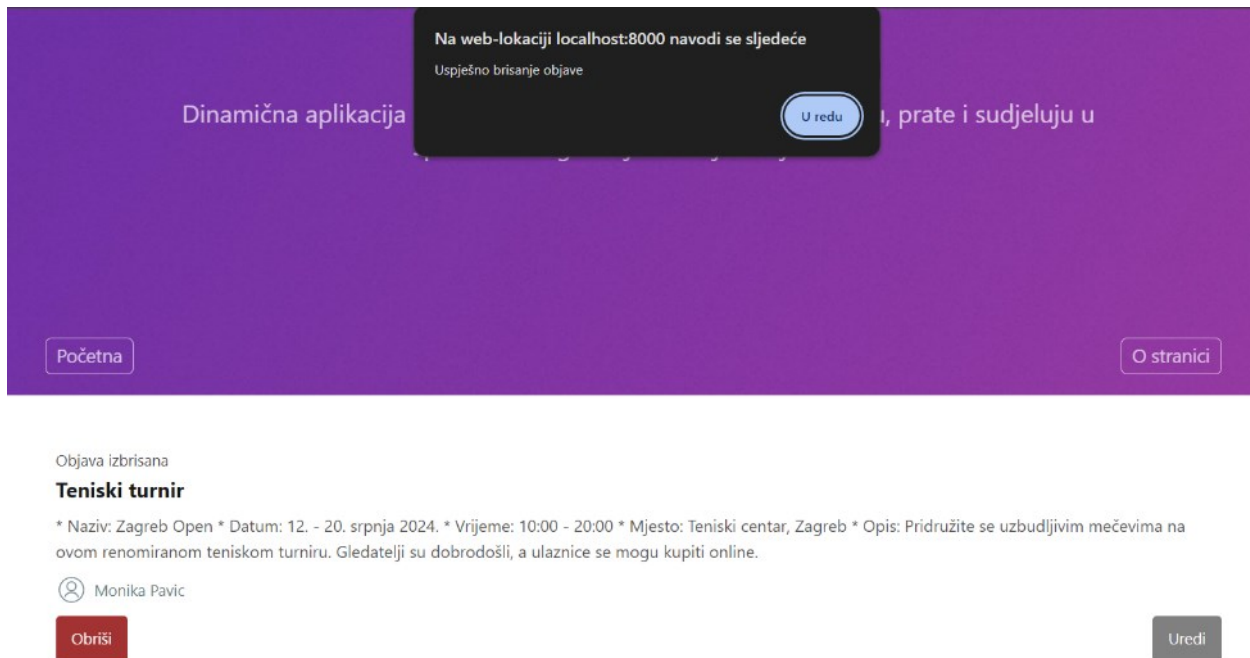
1. instalacija – `npm install react-quill`
2. dodavanje – pozicioniranje u datoteku u kojoj se nalazi komponenta, dodajemo uređivač i potrebne stilove:

```
import ReactQuill from 'react-quill';
```

```
import 'react-quill/dist/quill.snow.css';
```

3. implementacija – unutar datoteke u kojoj se dodaje uređivač, potrebno je pozicionirati se u dio koda gdje navedenu komponentu dodajemo

DELETE – brisanje objave – kreiran je gumb *Obriši* (slika 34.) na svakoj objavi čija je funkcija da omogući brisanje određene objave, odnosno u ovom slučaju aktivnosti. Kad se objava obriše javi se povratna poruka da je objava izbrisana, kao i prozor s potvrdom da je funkcija uspješno odrađena (slika 36.). Funkcija brisanja objave radi na način da se koristi JavaScript funkcija *fetch* za slanje zahtjeva prema Ghost CMS API-ju za brisanje određene objave; API URL koristi ID objave (postId) za identificiranje objave u Ghost CMS-u koju je potrebno obrisati, a za autorizaciju brisanja objave koristi se API admin ključ koji je generiran unutar Ghost CMS-a.



Slika 36. <https://github.com/mpavic2/diplomski/blob/master/src/templates/index.js>

Ono što je još važno napomenuti što se tiče same aplikacije je ažuriranje promjena koje se rade; ono što se radi unutar Gatsby projekta vidljivo je odmah na stranici, dok promjene koje se rade na Ghost administrativnom sučelju nisu vidljive dok se projekt ponovno ne pokrene naredbom `gatsby develop`.

8. RASPRAVA

Kao što je u prethodnim poglavljima objašnjeno, najvažnije karakteristike CMS tehnologije su jednostavnost korisničkog sučelja, upravljanje, ažuriranje, kreiranje sadržaja koji je odvojen od dizajna te multimedijaska podrška. Dok je samo Ghost administrativno sučelje jednostavno za korištenje kako je i opisano, problematika kreiranja aplikacije se pojavila kada je trebalo spojiti Ghost CMS i Gatsby. Praćenjem dokumentacije na službenoj Ghost stranici pri samoj instalaciji nije bilo nikakve greške, no Gatsby nije povlačio nikakve podatke iz Ghost CMS-a nego se na *localhostu* otvarao prikaz *gatsby default startera* koji ne uključuje Ghost. Putem naredbenog retka pokušano je ažuriranje dodataka i rješavanje upozorenja koja su bila vidljiva. Nakon svega pokušanog i istraženog, zaključak je da je problem bio u node.js verziji. Verzija koja je lokalno bila ažurirana je 18.x koja je po službenoj dokumentaciji podržana, no u praksi nije radila. Tek kad je verzija node.js-a ažurirana na 20.x, koja je ujedno i preporučena verzija za korištenje Ghost CMS-a, Gatsby i Ghost su se povezali, odnosno, Gatsby je povukao podatke iz Ghost CMS-a. Važno je napomenuti da nigdje prilikom instalacije i korištenja Ghost CMS-a nije bilo greške, radilo je uredno, samo kad se trebalo povezati s *frontend* Gatsby dijelom, tako da otkrivanje samo problema nije bilo jednostavno.

Prema službenim uputama, projekt se pokreće s naredbom

```
gatsby develop
```

koja je javljala error dok je instalirana verzija node.js-a bila 18.x i tad se projekt pokretao samo s naredbom

```
npm run develop
```

dok je na projektu koji je kreiran s instaliranom verzijom node.js 20.x bez problema pokretanje radilo s naredbom

```
gatsby develop
```

Ono što je olakšalo i pojednostavilo razvoj aplikacije je što su bilo kakve promjene unutar koda odmah vidljive, nema potrebe za osvježavanjem stranice ili pokretanjem projekta.

Na temelju napisanog možemo navesti prednosti i nedostatke implementacije Gatsby i Ghost CMS tehnologije.

PREDNOSTI

- Brzina i performanse
 - Ghost kao backend dio omogućava lagano kreiranje sadržaja, a zajedno s Gatsbyjevim brzim učitavanjem stranica poboljšava korisničko iskustvo
- Jednostavno održavanje
 - Ghost CMS administrativno sučelje omogućava upravljanje sadržajem bez potrebe pisanja koda
 - Promjene sadržaja na Gatsbyju i Ghostu su odmah vidljive što skraćuje vrijeme razvoja i olakšava održavanje aplikacije
- Jednostavna integracija s GraphQL-om
 - Gatsby koristi GraphQL za povlačenje podataka iz Ghost CMS-a što omogućuje fleksibilnost u načinu prikazivanja
- Sigurnost
 - Backend (Ghost) i frontend (Gatsby) dijelovi su odvojeni što smanjuje rizik sigurnosnih problema

NEDOSTACI

- Kompatibilnost
 - Podržana verzija Node.js-a nije povezala Ghost CMS i Gatsby, potrebno koristiti isključivo preporučenu verziju
 - Dokumentacija na službenim stranicama – nisu navedene poteškoće s verzijama
- Ograničena podrška lokalno
 - Ghost CMS administrativno sučelje podržava višejezičnost, ali je napisano samo na engleskom jeziku
 - Ručno prevođenje na hrvatski jezik produžilo razvoj aplikacije
- Dodatni alati

- Ako se CRUD komponente dodaju preko Gatsbyja, potrebno je dodati određene pakete i dodatke
- Ovisnost o vanjskom API-ju
 - Gatsby je ovisan o Ghost API-ju za dohvaćanje podataka

Ono što je potrebno još dodati je važnost dokumentacije; iako je u velikom dijelu razvoja aplikacije službena dokumentacija korisna za rješavanje problema, važno je i ažurirati istu, posebno kad se radi o novim verzijama i zastarjelim značajkama. No, suprotno tome, dokumentacija pruža smjernice za postavljanje, konfiguraciju i korištenje Ghost-a i Gatsbyja. Službena dokumentacija Gatsbyja pruža jasne korake kako integrirati Ghost i kako koristiti GraphQL upite za dohvaćanje podataka, dok je Ghost službena dokumentacija pomogla s razumijevanjem rada s Content API-jem i konfiguracijom Ghost instance.

U teorijskom dijelu navedena je arhitektura središta sadržaja gdje je navedeno da središte sadržaja centralizira sav sadržaj na jednom mjestu pomoću API-ja. u ovom slučaju središte sadržaja je Ghost CMS koji kao backend dio aplikacije omogućuje pohranu, upravljanje i isporuku sadržaja, a pomoću API ključa omogućuje prikaz, ažuriranje i brisanje sadržaja unutar aplikacije. Sadržaj koji je pohranjen u Ghost CMS-u je dostupan putem API-ja i može mu se pristupiti preko različitih platformi; preko mrežnih stranica, preko mobilne aplikacije ili nekog drugog kanala što omogućuje višekanalnu isporuku sadržaja.

Ta arhitektura središta sadržaja omogućuje odabir različitih alata i usluga; u primjeru aplikacije korištenje Gatsbyja kao frontend dijela aplikacije omogućilo je implementaciju WYSIWYG uređivača, odnosno react-quill paket za uređivanje sadržaja koji se koristi za uređivanje sportskih aktivnosti.

Projekt ukupno zauzima 1,15 GB memorije na disku sa svim dodacima i ovisnostima što je puno s obzirom da je kreirana jednostavna aplikacija u kojoj nije uključeno puno medijskih elemenata.

9. ZAKLJUČAK

Razvoj tehnologije i razvoj programskih jezika kroz godine nastoji olakšati i umanjiti prethodne poteškoće ili probleme na koje se nailazilo. Važno je pratiti dosadašnje trendove ali isto tako i prednosti i nedostatke pojedinih tehnologija kako bi razvoj išao u pravom smjeru. Tako se kroz godine razvijao i olakšavao razvoj mrežnih mjesta, nastankom pojedinih alata za automatizaciju i brojnih tehnologija kao što je i CMS. Dok alati za automatizaciju mrežnih mjesta ažuriraju sadržaj i funkcionalnost mrežnog mjesta, CMS tehnologija olakšava upravljanje i uređivanje sadržaja istog, a zajedno daju maksimalnu efikasnost u upravljanju mrežnih mjesta. Važno je napomenuti da iako takvi alati i tehnologija imaju puno prednosti kao što su jednostavnost i brzina korištenja, brza prilagodba, SEO funkcionalnost i primjena u različitim kontekstima kao što su blogovi, e – trgovine, poslovne stranice ali i edukativne platforme. postoje i neki nedostaci. Neki od nedostataka su sigurnosni rizici, posebno ako se ne ažurira, ograničena prilagodba u smislu nekakvih jedinstvenih potreba i funkcionalnosti, nekompatibilnost samih dodataka koji znaju uzrokovati problem s radom mrežnog mjesta, kao i troškovi održavanja *premium* verzija pojedinih alata. Kao što je prikazano u prethodnim poglavljima Aplikacija i Rasprava, postoje brojne prednosti ali i brojni nedostaci korištenja navedenih tehnologija; kao što je već opisano najveći problem bio je povezati Ghost i Gatsby zbog nepodržanih verzija. Kako bi se ti nedostaci izbjegli, naglasak treba biti na redovitom ažuriranju i održavanju sustava te korištenje sigurnosnih sustava. Ono što se može zaključiti iz prethodnih poglavlja da samo korištenje CMS sustava je jednostavno za korištenje, no problem je ažuriranje pojedinih dodataka i verzije koje nisu podržane, a isto tako i službena dokumentacija nije ažurirana i sve informacije koje su navedene nisu točne, što je dokazano u prethodnom poglavlju.

Zaključak cijelog rada je da praksa ne prati uvijek teoriju; navedeni sustavi bi trebali olakšati kreiranje aplikacija, no u ovom radu to nije bilo slučaj u potpunosti. Teorijski dio prikazao je CMS tehnologiju i Gatsby i Ghost CMS kao sustave koji olakšavaju razvoj mrežnih mjesta, ali praktični dio, može se reći, nije jednostavnošću pratio teorijski.

Gatsby i Ghost sustavi imaju potencijal za razvoj, kako zajedno, tako i odvojeno, unatoč nedostacima na koje se naišlo.

10.LITERATURA

1. Adobe Target. URL: <https://business.adobe.com/products/target/adobe-target.html>
2. Aspire Themes. Ghost themes for your blog, newsletter, podcast, portfolio, or news website. Accessible, simple, and easy to customize. URL: https://aspirethemes.com/?gclid=CjwKCAjwsvujBhAXEiwA_UXnANqRPQTUwtTth9VOSAKV4uQ_d7eOCwYOS2FdidREmSCCPGueezHQeRoCiNwQAvD_BwE
3. Evergage. URL: <https://www.salesforce.com/marketing/personalization/>
4. Gatsby. Documentation. Building with Components. URL: <https://www.gatsbyjs.com/docs/conceptual/building-with-components/>
5. Gatsby. Documentation. GraphQL & Gatsby. URL: <https://www.gatsbyjs.com/docs/graphql/>
6. Gatsby. Documentation. Layout Components. URL: <https://www.gatsbyjs.com/docs/how-to/routing/layout-components/>
7. Gatsby. Gatsby Starter Library. URL: <https://www.gatsbyjs.com/starters/gatsbyjs/gatsby-starter-hello-world>
8. Gatsby. Gatsby-plugin-styled-components. URL: <https://www.gatsbyjs.com/plugins/gatsby-plugin-styled-components>
9. Gatsby. Ghost. URL: <https://gatsby.ghost.org/about/>
10. Gatsby. How to Source Content from a Headless CMS. URL: <https://www.gatsbyjs.com/docs/how-to/sourcing-data/headless-cms/>
11. Gatsby. URL: <https://www.gatsbyjs.com/>
12. Gatsby. Welcome to the Gatsby Plugin Library. URL: <https://www.gatsbyjs.com/plugins>
13. Gatsby. Welcome to the Gatsby Way of Building. URL: <https://www.gatsbyjs.com/docs/>
14. Ghost. Developer docs. Translate. URL: <https://ghost.org/docs/themes/helpers/translate/>
15. Ghost. Developer docs. URL: <https://ghost.org/docs/>
16. Ghost. Ghost on the JAMstack. URL: <https://ghost.org/docs/jamstack/>
17. Ghost. How to install Ghost locally. URL: <https://ghost.org/docs/install/local/>
18. Ghost. Independent technology for modern publishing. URL: <https://ghost.org/>
19. Ghost. Working with Gatsby. URL: <https://ghost.org/docs/jamstack/gatsby/>

20. GhostCave. How to Translate Ghost CMS Members Portal into Your Language. URL: <https://ghostcave.org/how-to-translate-ghost-cms-members-portal-into-your-language/>
21. Github. Gatsby. URL: <https://github.com/gatsbyjs/gatsby>
22. Hackernoon. How to Build a Gatsby Static Site Using Ghost as a Headless CMS, 2021. URL: <https://hackernoon.com/how-to-build-a-gatsby-static-site-using-ghost-as-a-headless-cms-1k2o35sw>
23. History of content management systems and rise of headless CMS, 2023. URL: <https://www.contentstack.com/blog/all-about-headless/content-management-systems-history-and-headless-cms>
24. Medium. Why should I use GatsbyJS instead of WordPress and Ghost CMS for build my website, 2021. URL: <https://abdelhadi.medium.com/why-i-use-gatsby-for-my-site-instead-of-wordpress-and-ghost-cms-520895d6a836>
25. Monetate. URL: <https://monetate.com/>
26. Nodejs. An introduction to the npm package manager. URL: <https://nodejs.org/en/learn/getting-started/an-introduction-to-the-npm-package-manager>
27. Nodejs. Run JavaScript Everywhere. URL: <https://nodejs.org/en>
28. One Spot. URL: <https://www.onespotapps.com/>
29. Optimizely. URL: <https://www.optimizely.com/>
30. Salesforce Commerce Cloud. URL: <https://www.salesforce.com/commerce/>
31. Statcounter. Mobile market share worldwide. URL: <https://gs.statcounter.com/platform-market-share/desktop-mobile/worldwide/#yearly-2009-2024>
32. Strapi. Integrate Gatsby with Strapi. URL: <https://strapi.io/integrations/gatsby-cms>
33. Što je CMS. URL: <https://www.virtualna-tvornica.com/sto-je-cms/>
34. Što je CMS? Pregled sustava za upravljanje sadržajem i njihova primjena u izradi web stranica, 2024. URL: <https://copymate.app/hr/blog/multi/sto-je-cms-pregled-sustava-za-upravljanje-sadrzajem-i-njihova-primjena-u-izradi-web-stranica/>
35. Try Ghost. Ghost. URL: <https://github.com/TryGhost/Ghost/tree/main/ghost/i18n/locales?ref=ghostcave.org>