

Digitalna Kanban ploča

Maznik, Mando

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:776341>

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

Download date / Datum preuzimanja: **2024-11-23**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

Stručni studij Računarstvo

DIGITALNA KANBAN PLOČA

Završni rad

Mando Maznik

Osijek, 2024.

Obrazac Z1S: Obrazac za ocjenu završnog rada na stručnom prijediplomskom studiju

Ocjena završnog rada na stručnom prijediplomskom studiju

Ime i prezime pristupnika:	Mando Maznik
Studij, smjer:	Stručni prijediplomski studij Računarstvo
Mat. br. pristupnika, god.	AR 4794, 27.07.2020.
JMBAG:	0165085052
Mentor:	mr. sc. Željko Štanfel
Sumentor:	
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	doc. dr. sc. Ivana Hartmann Tolić
Član Povjerenstva 1:	mr. sc. Željko Štanfel
Član Povjerenstva 2:	Marina Peko, dipl. ing. el.
Naslov završnog rada:	%naziv_rada%
Znanstvena grana završnog rada:	Programsko inženjerstvo (zn. polje računarstvo)
Zadatak završnog rada:	Opiši što je Kanban ploča (Kanban Board) te proces korištenja ploče u Kanban metodologiji. Napiši jednostavnu aplikaciju koja prikazuje Kanban ploču s različitim statusima programskih zadataka na kojima radi tim programera. Omogući promjenu statusa zadatka od: ideja, planiran, u izradi, u testu, spreman za integraciju, integriran, napušten (ili sličan slijed). Zadatak može biti dodijeljen pojedincu ili timu ovisno o statusu zadatka. Pratimo broj dana za svaki zadatak u danom stanju. Zadatak koji je predugo u istom stanju signalizira tu situaciju. Demonstriraj aplikaciju.
Datum ocjene pismenog dijela završnog rada od strane mentora:	29.08.2024.
Ocjena pismenog dijela završnog rada od strane mentora:	Izvrstan (5)
Datum obrane završnog rada:	12.9.2024
Ocjena usmenog dijela završnog rada (obrane):	Izvrstan (5)
Ukupna ocjena završnog rada:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio stručni prijediplomski studij:	13.09.2024.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****IZJAVA O IZVORNOSTI RADA**

Osijek, 13.09.2024.

Ime i prezime Pristupnika:

Mando Maznik

Studij:

Stručni prijediplomski studij Računarstvo

Mat. br. Pristupnika, godina upisa:

AR 4794, 27.07.2020.

Turnitin podudaranje [%]:

4

Ovom izjavom izjavljujem da je rad pod nazivom: **Digitalna Kanban ploča**

izrađen pod vodstvom mentora mr. sc. Željko Štanfel

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

SADRŽAJ

1. UVOD	1
1.1. Zadatak završnog rada	1
2. KANBAN PLOČA	2
2.1. Svrha Kanban ploče.....	4
2.2. Ključne komponente Kanban ploče	5
2.2.1. Stupci.....	5
2.2.2. Kartice	5
2.2.3. Oznake i filteri	5
2.3. Implementacija Kanban ploče	6
2.3.1. Definiranje WIP granica	6
2.3.2. Analiza tijeka rada	7
2.3.3. Dizajn Kanban ploče	7
2.3.4. Razvoj i testiranje Kanban ploče	8
2.4. Kanban alati	8
2.4.1. Trello	8
2.4.2. Jira	10
2.4.3. Asana	11
2.5. Kanban i agilne metodologije.....	12
2.5.1. Kanban i Scrum	12
2.5.2. Kanban i Lean.....	13
2.6. Kanban u različitim sektorima	15
2.6.1. Kanban u proizvodnji	15
2.6.2. Kanban u zdravstvu	16
2.6.3. Kanban u obrazovanju	17
3. KORIŠTENE TEHNOLOGIJE	20
3.1. React.....	20
3.2. Firebase.....	22
4. RJEŠENJE	24
5. UPUTE ZA KORIŠTENJE APLIKACIJE	29
6. ZAKLJUČAK	35
LITERATURA	36

SAŽETAK.....	38
SUMMARY.....	39

1. UVOD

S obzirom na suvremeno poslovno okruženje u kojem je brzina i fleksibilnost od presudne važnosti, alati za upravljanje radnim procesima postaju neizostavni. Kanban ploča je jedan od najučinkovitijih vizualnih alata za upravljanje radnim procesima koja ujedno omogućava veću produktivnost tima. Ovaj jednostavni, a iznimno moćan sustav vizualizacijom pruža mogućnost praćenja procesa rada, prepoznavanja zastoja i optimizaciju procesa u stvarnom vremenu.

Kanban ploča sastoji se od stupaca koji označavaju različite faze rada, poput „Za napraviti“, „U procesu“, „Dovršeno“. Unutar svakog od tih stupaca postoje kartice koje prikazuju zasebne zadatke ili stavke rada. Ova vizualizacija pomaže timovima da jasno vide status svakog zadatka, prepoznaju gdje se pojavljuju zastoji i poduzmu potrebne korake kako bi poboljšali tijek rada.

Jedna od glavnih prednosti Kanban metode je njena fleksibilnost i prilagodljivost. Za razliku od tradicionalnih metoda upravljanja projektima, Kanban ne zahtijeva detaljno planiranje. Umjesto toga, omogućuje stalno prilagođavanje i usavršavanje procesa na temelju stvarnih podataka i povratnih informacija. Ova prilagodljivost čini Kanban izuzetno korisnim u dinamičnim okruženjima gdje su promjene česte i teško predvidljive.

Kanban metoda se ne koristi samo u poslovnom svijetu, već je mnogi ljudi koriste i u svakodnevnom životu, često nesvjesno. Osobe koje su preopterećene poslom često primjenjuju Kanban principe za organiziranje svojih zadataka. Na primjer, jednostavne liste zadataka na papiru ili u digitalnim aplikacijama služe kao osobne Kanban ploče. One pomažu pojedincima da jasno vide što treba obaviti, što je trenutno u tijeku i što je već završeno. Ovakav način organizacije može značajno smanjiti stres i povećati produktivnost, omogućujući bolju ravnotežu između poslovnih obaveza i osobnog života.

1.1. Zadatak završnog rada

Zadatak završnog rada je objasniti što je to Kanban ploča te na osnovnu dosadašnjeg stečenog znanja iz internet programiranja, napraviti jednostavnu aplikaciju koja prikazuje Kanban ploču s različitim statusima programskih zadataka na kojima radi tim programera.

2. KANBAN PLOČA

Kanban je spoj riječi s japanskim korijenima. Riječ „kan“ znači vizualno, dok „ban“ označava karticu. Kao uobičajeni japanski izraz, ove kombinirane riječi često se koriste za označavanje plakata ili vizualnih ploča [1].

Kanban metoda ima svoje početke u japanskoj industriji, posebno u proizvodnji, gdje je razvijena kao ključni dio Toyota Production System (TPS) ili Toyota Proizvodnog Sustava. Još u pedesetim godinama prošlog stoljeća Toyota je započela razvijati svoj proizvodni sustav pod vodstvom Taiichi Ohnoa i Eiji Toyodea. Kanban metoda nije samo nastala kao ključni dio Toyota Proizvodnog Sustava, već je tijekom godina prošla kroz različite faze razvoja. U početku se Kanban koristio kao jednostavan način za praćenje zaliha u proizvodnji, no s vremenom je postao sveobuhvatniji sustav za optimizaciju procesa i povećanje produktivnosti.



Sl. 2.1. Prostor za pohranu Kanban kartica [2]

Do 1980-ih, uspjeh Kanban sustava privukao je pažnju stručnjaka za proizvodnju i poslovnih lidera širom svijeta. Metoda je počela prelaziti granice Japana i proizvodne industrije, prilagođavajući se različitim sektorima. Tijekom ove tranzicije, koncept vizualnih signala za upravljanje radom i zalihama počeo se koristiti u upravljanju projektima i poslovnim procesima.

Krajem 1990-ih i početkom 2000-ih, kako je razvoj softvera postajao sve složeniji i zahtjevniji, programeri su počeli tražiti nove načine upravljanja projektima. U tom kontekstu, Kanban principi

su prepoznati kao učinkoviti alati za poboljšanje vidljivosti rada, smanjenje kaosa i optimizaciju timске suradnje. Programeri su počeli koristiti bijele ploče s karticama za vizualizaciju i dijeljenje statusa projekata, stvarajući strukturu koja je evoluirala u moderne Kanban ploče.



Sl. 2.2. Kanban ploča u fizičkom obliku [3]

Oko 2006. godine, kako je Kanban metoda postajala sve popularnija, programeri su počeli primjenjivati na uobičajenu praksu vizualizacije i dijeljenja statusa projekta postavljanjem kartica na bijele ploče u projektnoj sobi. Primjena Kanban principa na inače kaotične projektne ploče donijela im je stupce i strukturu iz kojih su se razvile moderne Kanban ploče. Sada se uspješno koriste u svim vrstama industrija i svugdje gdje vidljivost statusa rada pomaže u boljoj organizaciji posla [4].



Sl. 2.3. Kanban ploča u digitalnom obliku

2.1. Svrha Kanban ploče

Kanban pomaže timovima da poboljšaju način na koji razvijaju softver. Tim koji koristi Kanban ima jasnu sliku svih koraka koje poduzimaju u izradi softvera, kako komuniciraju s ostatkom tvrtke te prepoznaju gdje dolazi do gubitaka zbog neučinkovitosti i nepravilnosti. Koristeći Kanban, timovi mogu postupno poboljšavati svoje procese uklanjanjem glavnih uzroka tih problema. Poboljšanje procesa izrade softvera tradicionalno se naziva "procesno poboljšanje". Kanban je dobar primjer primjene agilnih ideja (kao što je donošenje odluka u posljednjem mogućem trenutku) kako bi se stvorila metoda za procesno poboljšanje koja je jednostavna i laka za usvajanje timovima [5].

Kanban ploča pomaže timovima da bolje upravljaju svojim vremenom i resursima. Kroz pravilo ograničenja rada u tijeku, timovi mogu kontrolirati broj zadataka koji se obavljaju istovremeno, što sprječava preopterećenje i omogućuje fokusiranje na dovršavanje postojećih zadataka prije nego što započnu nove. Ova praksa promiče disciplinu i pomaže u održavanju stalnog i održivog radnog tempa.

Osim toga, Kanban ploča potiče kontinuirano poboljšanje. Redovitim praćenjem i procjenom tijeka rada, timovi mogu identificirati obrasce i prepoznati prilike za poboljšanje procesa. Kroz povratne informacije i prilagodbe, timovi mogu neprestano usavršavati svoje radne metode, čineći cijeli proces učinkovitijim i fleksibilnijim prema promjenama.

Uvođenje Kanban ploča također može značajno podići motivaciju i angažman zaposlenika. Vidljivost napretka i postignuća na ploči može djelovati motivirajuće, jer članovi tima mogu jasno vidjeti svoj doprinos i razumjeti kako njihov rad utječe na ukupni projekt. Ovo može povećati osjećaj zadovoljstva i pripadnosti timu.

Kanban ploče su izuzetno fleksibilne i prilagodljive različitim potrebama i veličinama timova. Bez obzira na specifičnosti projekta ili industrije, Kanban se može prilagoditi kako bi odgovarao različitim radnim procesima i izazovima. Ova fleksibilnost čini Kanban ploče korisnim alatom za razne vrste projekata, od jednostavnih do kompleksnih.

2.2. Ključne komponente Kanban ploče

2.2.1. Stupci

Stupci su temeljni dijelovi Kanban ploče koji prikazuju različite faze rada. Svaki stupac predstavlja specifičan korak kroz koji zadatak prolazi od početka do završetka. Na primjer, standardni stupci uključuju „Ideja“, „U tijeku“ i „Završeno“. Ovi stupci omogućuju članovima tima da jasno vide status svakog zadatka i shvate u kojoj fazi se nalazi. Kroz upotrebu stupaca, timovi mogu lako uočiti zastoje u procesu i prepoznati faze koje trebaju dodatnu pažnju ili resurse.

2.2.2. Kartice

Kartice predstavljaju zadatke ili stavke koje su potrebne za obavljanje. Svaka kartica sadrži ključne informacije o zadatku, kao što su opis, osoba koja je odgovorna za njega, rokovi i relevantni dokumenti ili poveznice. Kartice se premještaju između stupaca kako zadaci prolaze kroz različite faze tijeka rada. Ovaj vizualni prikaz omogućuje članovima tima da brzo shvate što treba obaviti, tko je zadužen za koje zadatke i koji su prioritetni. Kartice također olakšavaju praćenje napretka i osiguravaju da se ništa ne zaboravi ili izgubi tijekom procesa.

2.2.3. Oznake i filteri

Oznake i filteri su dodatni alati koji pomažu u organizaciji i upravljanju zadacima na Kanban ploči. Oznake su poput vizualnih oznaka koje se koriste za sortiranje zadataka prema različitim kriterijima, poput vrste zadatka, prioriteta ili tima koji je odgovoran za njihovo izvršenje. Ove

oznake omogućuju timovima da brzo identificiraju određene vrste zadataka i imaju bolji pregled nad cijelim projektom.

S druge strane, filteri omogućuju korisnicima prilagodbu prikaza Kanban ploče prema specifičnim kriterijima, poput statusa zadatka, odgovorne osobe ili rokova. Na taj način, timovi mogu usmjeriti svoju pažnju na određene aspekte projekta i brže pronaći potrebne informacije. Filteri također olakšavaju praćenje napretka i identificiranje mogućih problema ili zastoja.

2.3. Implementacija Kanban ploče

Jednostavan opis koraka za završetak projekta temeljenog na Kanbanu uključuje analizu, dizajn, razvoj, testiranje i korištenje.

Prvi i najvažniji korak u implementaciji Kanban koncepta za poslovne procese je korištenje prednosti vizualnog upravljanja. Nakon završetka tog dijela procesa, sljedeći korak je postavljanje granica radova u tijeku ili WIP (engl. *Work In Progress*) granica [6].

2.3.1. Definiranje WIP granica

Definiranje WIP granica je ključni korak u implementaciji Kanban ploče. Ove granice definiraju maksimalni broj zadataka ili radnih jedinica koje tim može imati u isto vrijeme u procesu izvršavanja. Ideja je ograničiti rad u tijeku kako bi se spriječila preopterećenost i poboljšala učinkovitost.

Kroz postavljanje WIP granica, timovi dobivaju jasnu sliku o tome koliko posla mogu istovremeno obavljati. To pomaže u fokusiranju resursa na najvažnije zadatke, smanjuje višezadaćnost (eng. *multitasking*) i ubrzava vrijeme dovršetka pojedinih zadataka. Također, potiče timove da identificiraju i rješavaju prepreke koje mogu usporiti protok rada.

Bitno je istaknuti da postavljanje WIP granica nije fiksni proces. Timovi ih mogu prilagođavati ovisno o dinamici projekta, promjenama prioriteta ili drugim čimbenicima koji utječu na radne procese. Redovito praćenje i prilagođavanje WIP granica ključno je za održavanje agilnosti i kontinuirano poboljšanje u primjeni Kanbana.



Sl. 2.4. Primjer WIP limita

2.3.2. Analiza tijeka rada

U prvoj fazi analize tijeka rada, tim treba detaljno mapirati trenutne procese, te dokumentirati svaki korak koji se poduzima od početka do kraja projekta ili zadatka. Ova faza uključuje identifikaciju svih uloga i odgovornosti, komunikacijskih kanala, potrebnih resursa te vremena potrebnog za izvršenje svakog koraka.

Nakon mapiranja, tim nastavlja s procjenom efikasnosti i identifikacijom područja za poboljšanje. Kvalitetna analiza omogućuje otkrivanje slabosti, gubitaka i svih potencijalnih promjena koje bi trebalo napraviti. Iako neki sustavi koji uključuju strukturirano vođenje projekata mogu nositi određene rizike u implementaciji, Kanban se ističe minimalnim rizicima zbog svoje prilagodljivosti.

2.3.3. Dizajn Kanban ploče

Prvi korak u kreiranju Kanban ploče je postavljanje stupaca koji će prikazivati različite faze radnog procesa. Važno je da stupci vjerno prikazuju stvarne faze kroz koje zadaci prolaze. Kanban ploča također treba omogućiti praćenje i analizu učinka. Korištenje vizualnih indikatora poput boja ili naljepnica može olakšati brže prepoznavanje prioriteta, ključnih zadataka ili potencijalnih problema. Redovita analiza ploče i prilagođavanje dizajna na temelju prikupljenih podataka neophodni su za stalno poboljšanje procesa. Jedna od najvećih pogrešaka prilikom mapiranja projekta i organiziranja procesa je vizualiziranje idealnog projekta umjesto stvarnog tijeka rada.

2.3.4. Razvoj i testiranje Kanban ploče

Razvoj Kanban ploče je kontinuirani proces, a ne jednokratni zadatak. Pri kreiranju Kanban kartica i ploča, važno je pažljivo razmotriti što svaka od njih treba sadržavati. Redovito praćenje i analiza ploče su ključni za identificiranje područja koja zahtijevaju poboljšanja te važno je primijeniti minimalno opterećenje kako bi se osiguralo da timovi ne budu preopterećeni. Timovi moraju biti spremni prilagoditi dizajn ploče na temelju prikupljenih podataka i povratnih informacija kako bi osigurali kontinuirano unapređenje radnih procesa.

Testiranje Kanban ploče predstavlja ključni korak u njejoj implementaciji jer osigurava da sustav radi učinkovito i da je prilagodljiv potrebama tima. Ovaj proces omogućava timovima da identificiraju i uklone moguće nedostatke, optimiziraju radne tokove te osiguraju stalno poboljšavanje. Ključni aspekt testiranja Kanban ploče je prikupljanje povratnih informacija od svih članova tima. Na temelju tih podataka, tim provodi detaljnu analizu učinkovitosti Kanban ploče. Isto tako od velike važnosti je pronaći WIP granicu koja će odgovarati timu. Moramo paziti da granica ne bude previsoka ili premalena. Stoga, pronalaženje ravnoteže zahtijeva pažljivu analizu i razvoj. Prilagodbe ploče mogu uključivati promjene u dizajnu stupaca, redefiniranje WIP granica ili modifikaciju kartica kako bi bolje odgovarale potrebama tima.

2.4. Kanban alati

Trello, Jira i Asana su tri popularna alata za Kanban, svaki s vlastitim karakteristikama, prednostima i nedostacima. Trello je pogodan za jednostavne projekte i timove koji žele brzo vizualizirati svoj rad, dok Jira nudi napredne mogućnosti i fleksibilnost potrebnu za složene IT projekte. Asana se nalazi između ova dva, pružajući jednostavno, ali snažno rješenje za različite timove. Odabir odgovarajućeg alata ovisi o specifičnim potrebama projekta i tima, a razumijevanje njihovih razlika može biti ključno za uspješnu primjenu Kanban metodologije.

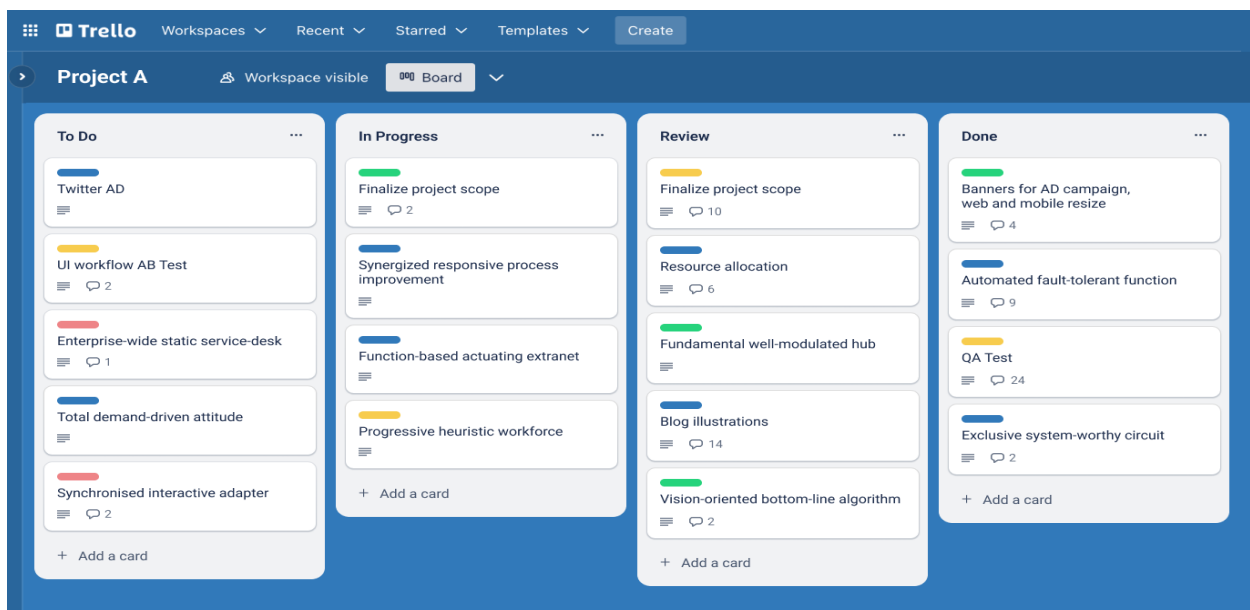
2.4.1. Trello

Trello je idealna platforma za Kanban timove. Možete lako konfigurirati ploču kako bi odgovarala vašem procesu, a ona je vidljiva svugdje, čak i kada vaš tim radi potpuno ili djelomično na daljinu [7].

Jedan je od najjednostavnijih i najintuitivnijih Kanban alata, omiljen među mnogim timovima zbog svoje pristupačnosti i fleksibilnosti. Korisnici mogu kreirati ploče koje sadrže liste i kartice, omogućujući vizualizaciju zadataka i njihovog napretka.

Jedna od glavnih prednosti Trelle je njegova jednostavna upotreba. Zahvaljujući intuitivnom sučelju, korisnici ga brzo usvajaju i lako koriste. Mogu jednostavno premještati kartice između stupaca kako bi ažurirali status zadataka. Trello također omogućava integracije s drugim alatima poput Slacka, Google Drive i GitHub, što korisnicima omogućuje povezivanje različitih aspekata njihovog rada. Trello se lako prilagođava različitim potrebama projekata, bilo da su osobne ili poslovne prirode. Besplatna osnovna verzija Trelle je dostupna, s mogućnošću nadogradnje na plaćene verzije koje nude dodatne funkcionalnosti.

Ipak, Trello ima ograničene napredne mogućnosti koje su dostupne samo u plaćenim verzijama, što može predstavljati nedostatak za korisnike koji traže detaljnije analize i izvještavanje. Nadalje, za vrlo velike ili kompleksne projekte gdje je potrebna veća skalabilnost, Trello može biti manje učinkovit.



Sl 2.5. Primjer Trello Kanban ploče [8]

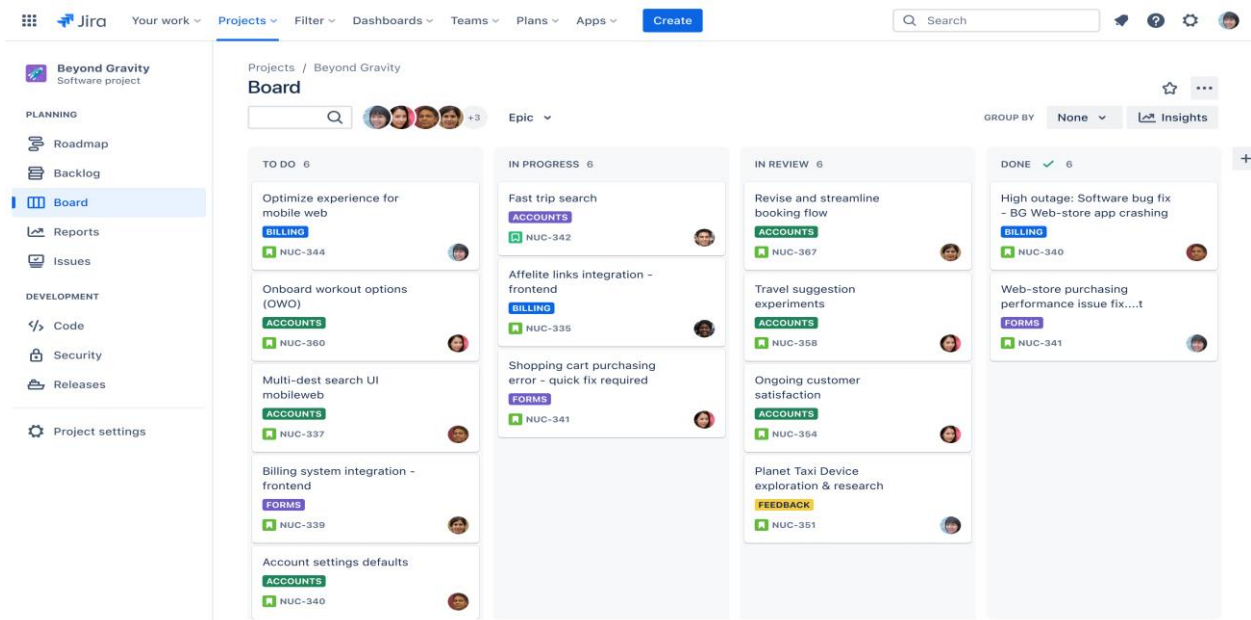
2.4.2. Jira

Jira je razvijena od strane Atlassian, alat je za upravljanje projektima koji je često korišten u IT sektoru, posebno za razvoj softvera. Podržava Kanban, Scrum i ostale agilne metode, čineći je izuzetno prilagodljivom za različite vrste timova i projekata.

Jira nudi Kanban ploču, gdje se puno više pridržavaju originalne ideje. Umjesto jednostavnih kartica, svaka stavka može biti konfigurirana s prilagodljivim poljima. Iako biste mogli pokušati stvarati vlastite elemente, preimenovati stupce i općenito gurati granice, nagrađuje strukturiraniji i formalniji pristup elementima. Cijelo korisničko iskustvo potiče vas da stvorite zadanu strukturu stavki koja je specifična za potrebe vašeg tima i projekta [9].

Jedna od glavnih prednosti Jira je njegova sposobnost za napredno praćenje zadataka. Jira omogućuje detaljno nadgledanje zadataka, uključujući prioritete, rokove i povezane zadatke. Osim toga, pruža snažne funkcionalnosti za analitiku i izvještavanje, omogućujući detaljne uvide u performanse tima i projekata. Integracije s alatima kao što su Confluence, Bitbucket i GitHub dodatno poboljšavaju suradnju i tok rada.

Ipak, Jira može biti izazovan alat za nove korisnike, s prilično zahtjevnim procesom učenja. Nadalje, Jira je skuplja opcija, posebno za veće timove koji zahtijevaju napredne funkcionalnosti.



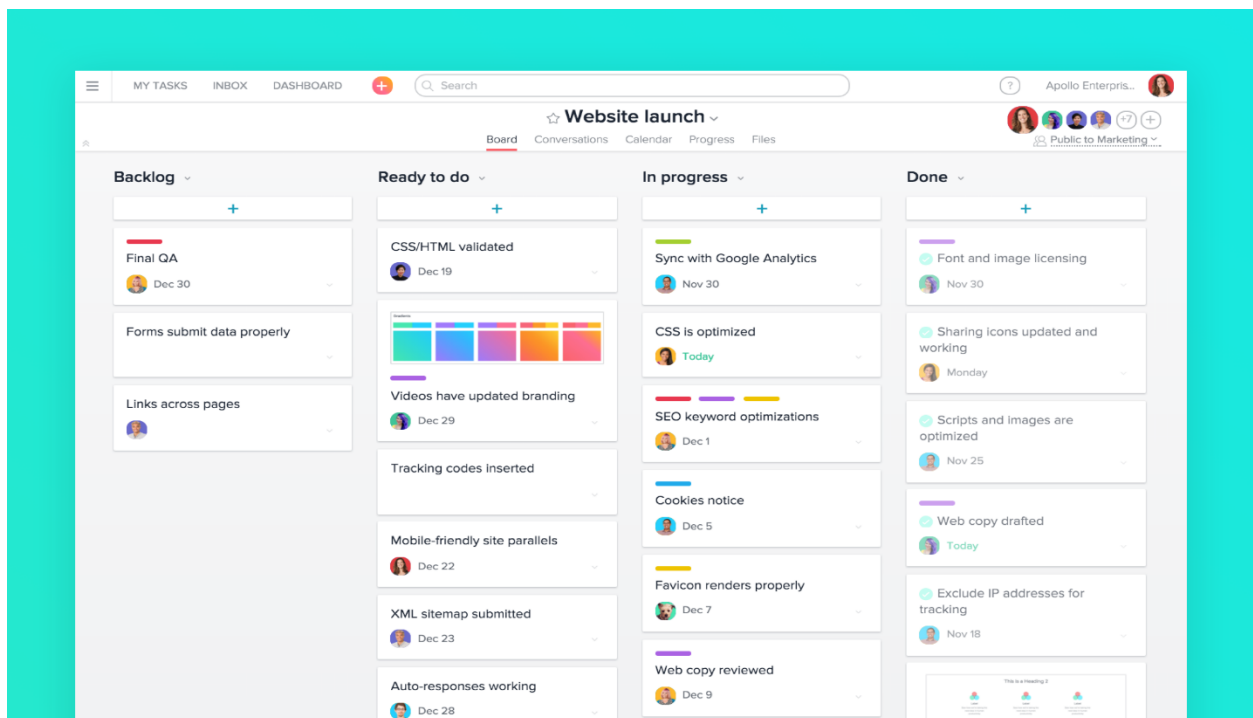
Sl. 2.6. Primjer Jira Kanban ploče [10]

2.4.3. Asana

Asana, poznati alat za upravljanje projektima, koristi tehniku ploče Kanban kako bi pomogao timovima da lakše prepoznaju i prate projekte. Asana Kanban pomaže timovima da izbjegnu zabunu o napretku zadatka i sprječava nesporazume među članovima ili odjelima. Ova ploča se automatski ažurira u stvarnom vremenu, tako da timovi mogu odmah vidjeti promjene. Asana Kanban također korisnicima omogućuje dodjeljivanje rokova, dodavanje podzadataka unutar svake kartice i ažuriranje komentara kada članovi surađuju na platformi [11].

Integracije s alatima kao što su Slack, Google Drive i Microsoft Teams omogućavaju povezanost radnih procesa. Asana je poznata po svojoj jednostavnosti korištenja i intuitivnom sučelju, što je čini pristupačnom za timove različitih veličina i potreba.

Ipak, napredne funkcionalnosti su dostupne samo u plaćenim verzijama, što može biti ograničavajuće za korisnike koji traže besplatna rješenja. Nadalje, Asana može biti manje učinkovita za vrlo velike ili složene projekte, gdje je potrebna veća skalabilnost.



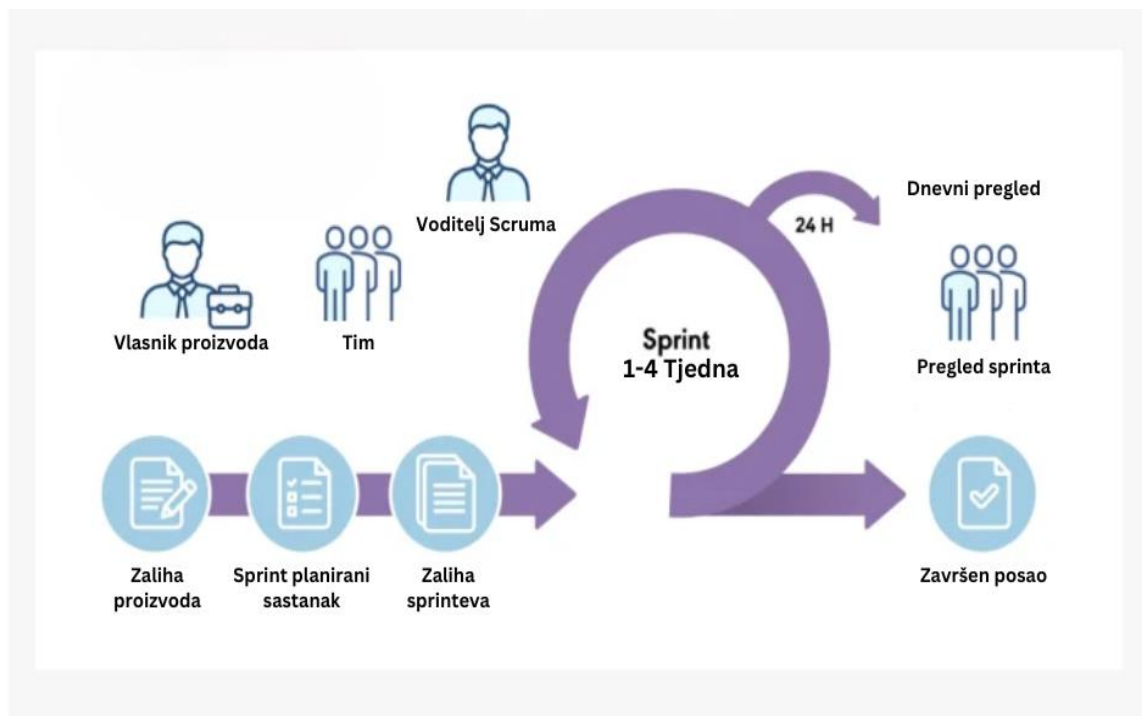
Sl. 2.7. Primjer Asana Kanban ploče [12]

2.5. Kanban i agilne metodologije

2.5.1. Kanban i Scrum

Kanban i Scrum su dvije popularne agilne metodologije koje se koriste za upravljanje projektima i razvoj softvera. Unatoč različitim pristupima, moguće ih je kombinirati kako bi se postigla optimalna agilna praksa.

Scrum je okvir za upravljanje projektima koji koristi iterativni i inkrementalni pristup razvoju. Timovi rade unutar fiksnih vremenskih intervala poznatih kao *sprint*, koji obično traju između dva i četiri tjedna. Svaki *sprint* započinje planiranjem tijekom kojeg tim definira ciljeve i zadatke za nadolazeći period. Na kraju *sprint*-a, održava se pregled kako bi se procijenio postignuti napredak, a retrospektiva kako bi se identificirala područja za poboljšanje. U Scrum timovima postoje jasno definirane uloge, uključujući voditelj scruma (eng. *Scrum master*), vlasnik proizvoda (eng. *Product owner*) i razvojni tim (eng. *Team*), što pomaže u strukturiranju rada i raspodjeli odgovornosti.



Sl. 2.8.. Scrum proces [13]

Iako Kanban i Scrum dijele zajedničke agilne principe, kao što su fleksibilnost, kontinuirano poboljšanje i prilagodljivost, oni se razlikuju u pristupu i načinu primjene. Obje metodologije

koriste vizualne alate za praćenje rada, što je jedna od njihovih glavnih sličnosti. Korištenje ploča za prikaz trenutnog statusa zadataka omogućuje timovima bolju transparentnost i preglednost radnih procesa. Ova vizualna komponenta pomaže u prepoznavanju zastoja i omogućava pravovremenu intervenciju kako bi se optimiziralo izvršenje zadataka.

Ključna razlika između Kanbana i Scrum-a leži u strukturi i fleksibilnosti. Scrum se ističe svojom strukturom koja uključuje fiksne vremenske okvire poznate kao *sprint*, što omogućuje fokus na kratkoročne ciljeve i kontinuirano evaluiraju svoj napredak. Međutim, fiksni vremenski okviri mogu biti ograničavajući za zadatke koji ne odgovaraju točno tim intervalima. S druge strane, Kanban je znatno fleksibilniji jer nema fiksne vremenske okvire. Umjesto toga, koristi granice radova u tijeku, odnosno WIP granice kako bi ograničio preopterećenje tima, što omogućuje kontinuirani rad i prilagodbu u realnom vremenu.

Također, razlika je u definiranju uloga unutar tima. Scrum ima jasno definirane uloge poput *Scrum Master*, *Product owner* i *Team*, što pomaže u strukturiranju rada i jasnom raspoređivanju odgovornosti. Nasuprot tome, Kanban nema strogo definirane uloge, što može pružiti veću fleksibilnost, ali istovremeno zahtijeva visoku razinu samoupravljanja tima.

Moguće je kombinirati Kanban i Scrum kako bi se iskoristile prednosti oba pristupa. Na primjer, timovi mogu primijeniti Scrum za strukturirane *sprint*-ove i redovite evaluacije, istovremeno koristeći Kanban za upravljanje radom u tijeku i prilagodbu promjenama. Ova kombinacija može doprinijeti većoj agilnosti tima, poboljšati transparentnost i optimizirati radne procese.

Kod Kanbana i Scrum-a, timovi sami biraju kako će upravljati delegacijama unutar svojih malih grupa. Njihova dužnost i misija je održavati efikasan tok vrijednosti prema svojim klijentima. Time mogu brzo reagirati na promjenjive okolnosti dok istovremeno razvijaju konkurentne proizvode i usluge unutar razumnih budžeta [14].

2.5.2. Kanban i Lean

Kanban i Lean su dva upravljačka pristupa koji su nastali s ciljem unapređenja produktivnosti, smanjenja gubitaka i poboljšanja kvalitete u proizvodnim i poslovnim procesima. Iako se koriste samostalno, njihova kombinacija može stvoriti snažan alat za kontinuirano poboljšanje.



Sl. 2.9. Lean proces [15]

Lean je agilna metodologija koja poboljšava isporuku proizvoda smanjenjem otpada (engl. *garbage*), te je vrlo učinkovita. Definicija otpada je prilično drugačija u Lean terminima, pa se sve što ne dodaje nove funkcionalnosti konačnom proizvodu smatra otpadom. Stoga ovaj pristup čini Lean agilnim jer postoji potrebna iterativna struktura projekata. Mnogo je lakše ukloniti otpad tijekom iterativnog ciklusa nego ga ukloniti kada je proizvod spreman [16].

Kanban i Lean dijele nekoliko zajedničkih ciljeva i vrijednosti, poput brze i česte isporuke vrijednosti kupcima, smanjenja otpada i neučinkovitosti u procesu, unapređenje kvalitete i zadovoljstva kupaca, poticanje timova na donošenje odluka i rješavanje problema te prilagodbu promjenjivim zahtjevima i povratnim informacijama. Ovi ciljevi su ključni za postizanje uspjeha u razvoju proizvoda ili pružanju usluga.

Kanban i Lean se razlikuju u nekoliko ključnih aspekata, uključujući fleksibilnost i propisnost. Kanban je manje strog i pruža veću fleksibilnost od Leana jer ne zahtijeva specifične uloge, procese ili alate. Također, Kanban je usredotočeniji na protok rada, dok Lean više naglašava vrijednost rada. Kanban je bolji izbor za složen i nepredvidiv rad, dok je Lean pogodniji za jednostavan i stabilan rad. Kanban može lakše nositi s promjenjivim prioritetima i ovisnostima, dok Lean zahtijeva jasan i stabilan tok vrijednosti. Metrike poput vremena ciklusa, propusnosti, vremena do isporuke, vrijednosti za kupca, otpada i kvalitete mjere se na različite načine između ova dva pristupa.

Primjenom Lean principa kao što su „just-in-time“ (pravovremena proizvodnja) i „jidoka“ (automatsko otkrivanje problema), Kanban postaje još učinkovitiji alat za optimizaciju procesa. Ukratko, Kanban i Lean su metodologije koje se međusobno nadopunjuju i mogu se koristiti zajedno kako bi se postigla veća učinkovitost, smanjili gubici i kontinuirano unapređivala radna okolina. Integracija Kanbana i Lean principa rezultira agilnijim, fleksibilnijim i produktivnijim poslovnim procesima.

2.6. Kanban u različitim sektorima

2.6.1. Kanban u proizvodnji

Korištenje Kanbana u proizvodnji može pomoći u održavanju optimalnih razina zaliha proizvodnje. To se postiže signalizacijom kada je potrebno proizvesti ili naručiti zalihe. Na taj način, razine zaliha nikada nisu previsoke niti preniske. Ovo donosi višestruke prednosti, poput uštede novca, poboljšanja novčanog toka i smanjenja otpada pomoću Kanbana u proizvodnji. Time se smanjuje pretjerano skladištenje artikala, što povećava troškove skladištenja, ili nedovoljno skladištenje, što može poremetiti proizvodnju [17].

Osim toga, Kanban omogućuje efikasnije praćenje procesa proizvodnje. Korištenjem vizualnih signala, timovi mogu brzo identificirati barijere i područja koja se mogu unaprijediti, što rezultira većom učinkovitošću i produktivnošću. Kanban također olakšava uspostavljanje ritma proizvodnje, poznatog kao „pull“ sustav, gdje se novi proizvodi pokreću prema stvarnoj potražnji, a ne prema predviđanjima.

Uvođenje Kanbana može unaprijediti komunikaciju i suradnju unutar proizvodnog tima. Vizualne prezentacije toka rada omogućuju svim članovima tima jasno praćenje aktivnosti u tijeku, identificiranje zadataka koji su dovršeni te određivanje prioriteta. Ovo smanjuje mogućnost nesporazuma i povećava transparentnost.

Implementacija Kanbana može također podići zadovoljstvo zaposlenika. Definiranje uloga i očekivanja, zajedno s transparentnim praćenjem napretka, stvara radno okruženje u kojem su zaposlenici motiviraniji i angažiraniji. To rezultira većom produktivnošću i smanjenjem odlazaka zaposlenika, što dodatno pridonosi uspjehu proizvodnih poduzeća.

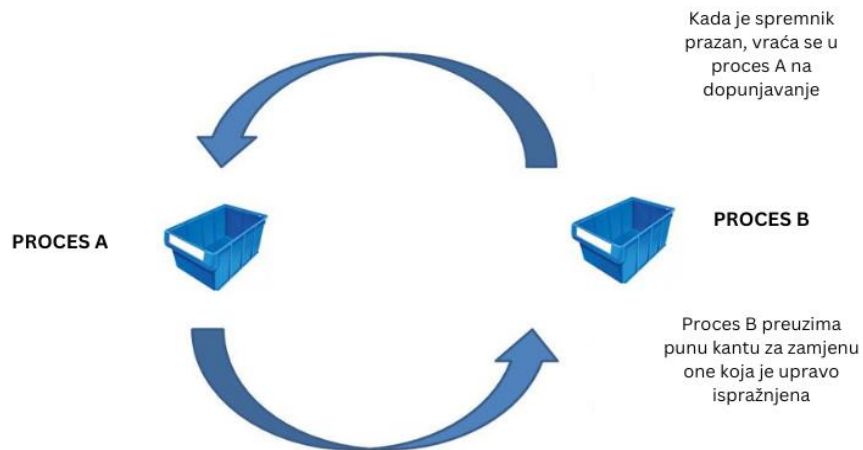


Sl. 2.10. Kanban u proizvodnoj industriji [18]

2.6.2. Kanban u zdravstvu

U današnjem brzo mijenjajućem zdravstvenom okruženju, učinkovite strategije upravljanja ključne su za osiguranje učinkovite i visokokvalitetne skrbi pacijenata. Jedna takva strategija koja postaje popularna je Kanban, vizualni alat za upravljanje koji je prvotno razvijen u industriji proizvodnje. Kanban pruža zdravstvenim organizacijama sistematičan i fleksibilan pristup kako bi optimizirale procese, smanjile gubitke i poboljšale ishode pacijenata [19].

Korištenjem *2-bin* Kanban sustava u zdravstvu predstavlja praktičan pristup za upravljanje zalihama medicinskih materijala. Ovaj sustav se oslanja na ideju korištenja dvije kutije (eng. *bin*) za svaki medicinski materijal - jedna koja je puna, a druga prazna. Kada se potroši materijal iz prve kutije, to služi kao signal za opskrbu jer se prazna kutija zamjenjuje punom, osiguravajući kontinuiranu dostupnost potrebnih materijala.



Sl. 2.11. 2-bin Kanban proces

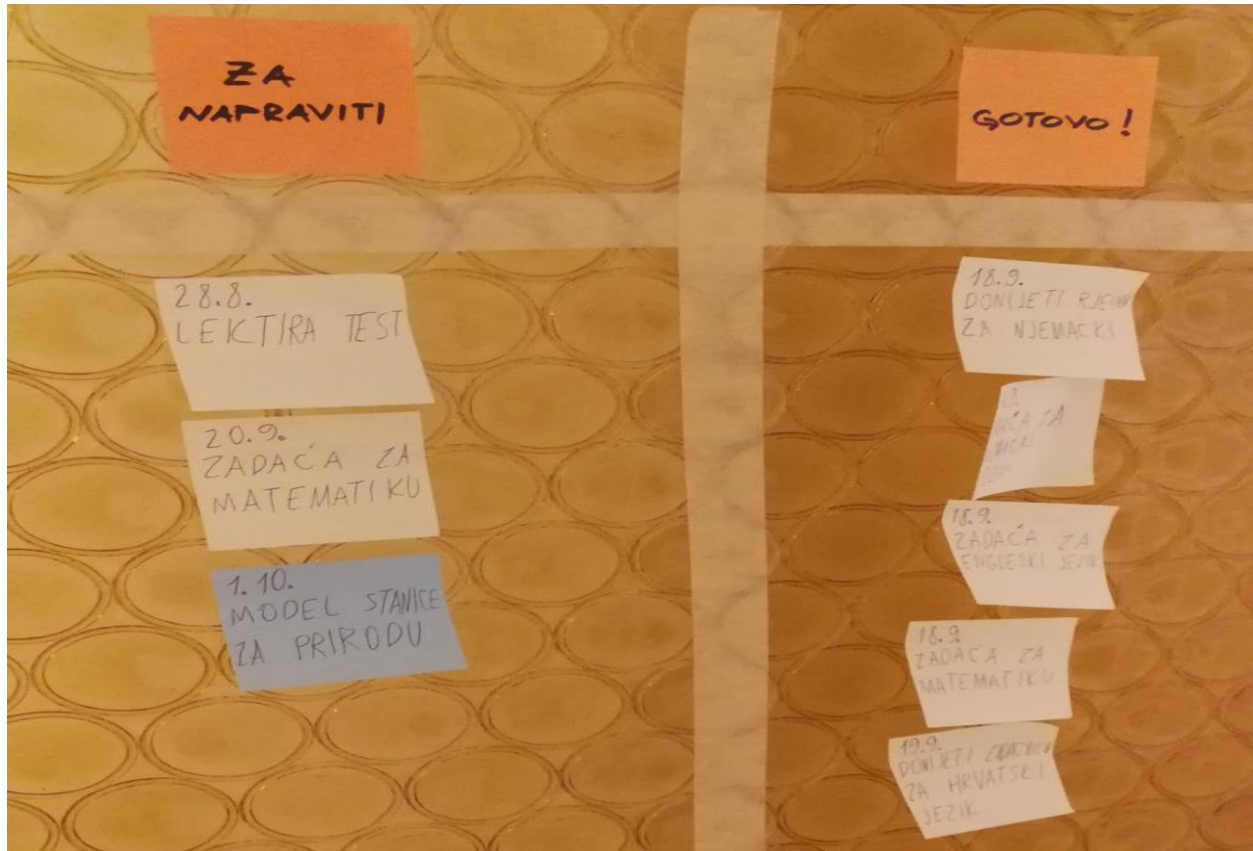
Primjena 2-bin Kanban sustava u zdravstvenim ustanovama omogućava učinkovito upravljanje zalihama, smanjujući rizik od prekomjernih zaliha ili nestašica jer se opskrba temelji na stvarnim potrebama i potrošnji.

Osim toga, ovaj sustav olakšava praćenje potrošnje materijala i identificiranje potencijalnih problema u opskrbnom lancu. Zamjena prazne kutije punom jasno signalizira da je materijal potrošen i da je potrebno ponovno opskrbiti zalihe. To omogućava zdravstvenim ustanovama da budu proaktivne u upravljanju zalihama i osiguraju neprekidan protok potrebnih materijala za pružanje visokokvalitetne zdravstvene skrbi.

2.6.3. Kanban u obrazovanju

Uvođenje Kanban ploča u obrazovne ustanove mijenja način na koji roditelji i učitelji međusobno komuniciraju i surađuju u vezi s obrazovnim putovanjem djeteta. Pružajući jasan, vizualni prikaz zadataka, domaćih zadaća i napretka, Kanban ploče omogućuju roditeljima da budu informirani o školskim aktivnostima svog djeteta, razumiju trenutnu fazu njihovog učenja i prepoznaju područja gdje je potrebna dodatna podrška. Ova otvorena linija komunikacije osigurava da učitelji i roditelji mogu učinkovitije raditi zajedno, stvarajući partnerstvo utemeljeno na transparentnosti i međusobnom povjerenju. Takvi zajednički naponi ključni su za stvaranje poticajnog okruženja za

učenje koje odgovara potrebama učenika, potičući njihov rast i razvoj, kako akademski, tako i osobno [20].



Sl. 2.12. Primjer Kanban ploče u osnovnoškolskom obrazovanju [21]

Primjena Kanban ploča u obrazovnim ustanovama ne samo da poboljšava komunikaciju između roditelja i učitelja, već također značajno unapređuje učinkovitost i angažman učenika. Vizualizacija zadataka i napretka omogućuje učenicima da jasnije vide svoje obaveze i ciljeve, što im pomaže u boljoj organizaciji vremena i postavljanju prioriteta. Kroz praćenje svog napretka na Kanban ploči, učenici razvijaju osjećaj odgovornosti i samodiscipline, što su ključne vještine za njihov budući akademski i profesionalni uspjeh.

Implementacija Kanban ploča može također potaknuti suradnju među učenicima. Grupni zadaci i projekti mogu biti organizirani i praćeni putem Kanban ploča, čime se potiče timski rad i zajedničko rješavanje problema. Učenici uče kako surađivati, dijeliti ideje i zajednički raditi na postizanju ciljeva, što su vještine koje će im biti od velike koristi u budućem obrazovanju i karijeri.

Kanban ploče mogu biti digitalne ili fizičke, što omogućuje njihovu prilagodbu različitim obrazovnim okruženjima. Digitalne Kanban ploče, integrirane s drugim alatima za upravljanje učenjem, omogućuju lako praćenje i dijeljenje informacija među učiteljima, učenicima i roditeljima. Ova integracija tehnologije dodatno povećava transparentnost i učinkovitost obrazovnog procesa.

3. KORIŠTENE TEHNOLOGIJE

3.1. React

React je okvir koji koristi Webpack za automatsko prevođenje React, JSX i ES6 koda, dok istovremeno obrađuje prefikse CSS datoteka. React je biblioteka za razvoj korisničkog sučelja temeljena na JavaScriptu. Iako je React biblioteka, a ne jezik, široko je korišten u web razvoju. Biblioteka se prvi put pojavila u svibnju 2013. godine i sada je jedna od najčešće korištenih frontend biblioteka za web razvoj [22].

Jedna od glavnih prednosti Reacta je njegova komponentna struktura, koja omogućuje programerima da podijele korisničko sučelje na manje, ponovno iskoristive dijelove. Svaka komponenta može imati svoju vlastitu logiku i stanje, što čini upravljanje složenim aplikacijama i održavanje koda jednostavnijim. Uz to, React koristi virtualni DOM (engl. *Document Object Model*), koji znatno poboljšava performanse aplikacija tako što učinkovitije upravlja stvarnim DOM-om.

DOM je strukturirani prikaz HTML-a na web stranici ili u aplikaciji. Predstavlja cjelokupno korisničko sučelje web aplikacije kao strukturu stabla podataka. Kad god dođe do promjene stanja korisničkog sučelja aplikacije, DOM se ažurira i prikazuje tu promjenu. DOM se ponovno prikazuje i manipulira sa svakom promjenom kako bi se ažuriralo korisničko sučelje aplikacije, što utječe na performanse i usporava rad [23].

React također pruža podršku za renderiranje na strani poslužitelja (eng. *Server Side Rendering*), što može poboljšati performanse aplikacija i optimizirati ih za tražilice. Ovo je posebno korisno za aplikacije koje trebaju biti lako dostupne i vidljive putem pretraživača.

Postoji mnogo dodataka i biblioteka koje proširuju mogućnosti Reacta, poput React Routera za upravljanje rutama unutar aplikacija i Reduxa za upravljanje stanjem aplikacije. Ove dodatne biblioteke omogućuju programerima da izgrade robusne i skalabilne aplikacije.

React definirani povratni pozivi funkcija (eng. *React hooks*) potpuno su transformirali način na koji programeri upravljaju stanjem i ostalim React funkcionalnostima unutar funkcionalnih komponenti. Prije nego što su React povratni pozivi uvedeni, upravljanje stanjem i životnim ciklusima komponenti zahtijevalo je korištenje klasa, što je često rezultiralo složenim i teško održivim kodom. Uvođenjem povratnih poziva, ovaj proces postao je mnogo jednostavniji.

Jedan od najpopularnijih povratnih poziva je *useState*, koji omogućava dodavanje stanja u funkcionalne komponente. Korištenjem *useState*, programeri mogu lako inicijalizirati i ažurirati stanje, čineći komponente dinamičnijima i interaktivnijima. Drugi bitan povratni poziv je *useEffect*, koji se koristi za izvođenje nuspojava kao što su dohvaćanje podataka, manipulacija DOM-om i postavljanje pretplatnika. *useEffect* omogućava jednostavno upravljanje životnim ciklusima komponenti, što čini kod lakšim za održavanje i razumijevanje.

Zbog svih ovih razloga, React ostaje prvi izbor za mnoge web programere kada se radi o razvoju modernih, interaktivnih i visokokvalitetnih web aplikacija. Bez obzira na to je li netko početnik ili iskusni programer, učenje i korištenje Reacta može značajno unaprijediti njegove vještine i mogućnosti u svijetu web razvoja.

```
JS App.js M X
src > JS App.js > App
1  import logo from './logo.svg';
2  import './App.css';
3
4  function App() {
5    return (
6      <>
7      <h1>Hello World</h1>
8      </>
9    );
10 }
11
12 export default App;
13
```

Sl. 3.1. Primjer React koda

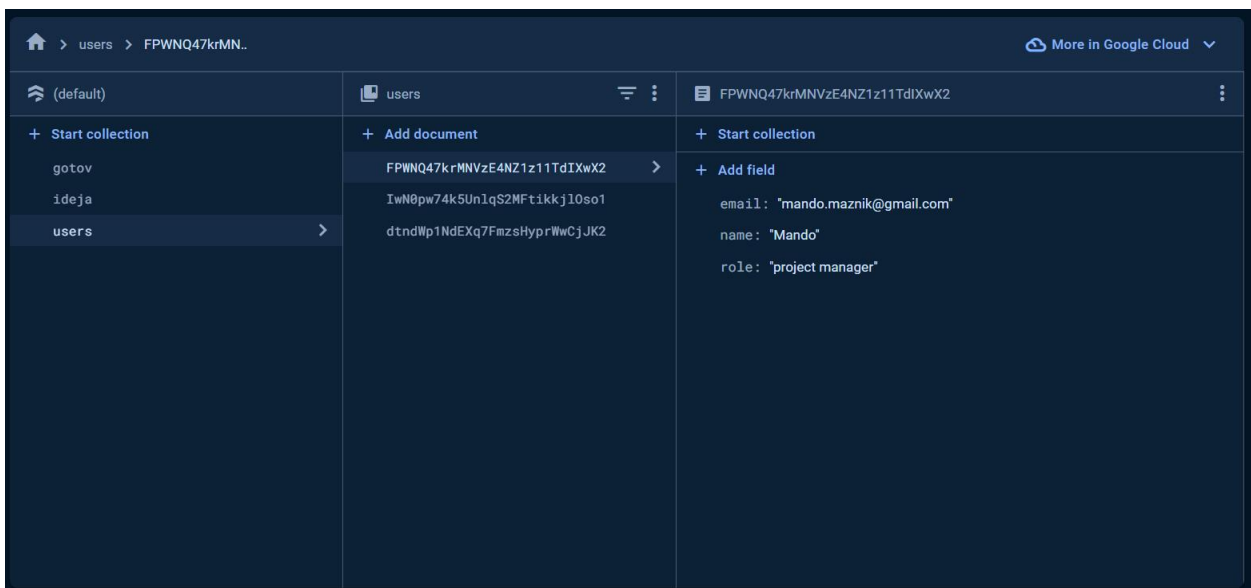
Na slici 3.1. može se vidjeti jednostavan primjer React koda. U tom primjeru se prikazuje kako se na ekran ispisuje riječ „Hello World“.

3.2. Firebase

Firebase je proizvod tvrtke Google koji pomaže developerima u bržoj, sigurnijoj i jednostavnijoj izradi, upravljanju i rastu njihovih aplikacija. Developerima omogućava bržu izradu aplikacija bez potrebe za programiranjem na samoj platformi Firebase, što olakšava efikasno korištenje njegovih značajki. Firebase pruža usluge za Android, iOS, web i Unity aplikacije te omogućava pohranu podataka u oblaku. Za pohranu podataka koristi NoSQL bazu podataka [24].

Za potrebe ovog projekta koristi se Firebase Firestore i Firebase *Authentication*, dvije ključne usluge iz Firebase platforme, kako bi se osiguralo učinkovito upravljanje podacima i autentifikaciju korisnika.

Firestore je fleksibilna, skalabilna baza podataka za razvoj mobilnih, web i serverskih aplikacija, koju pružaju Firebase i Google Cloud. Firestore održava sinkronizaciju podataka između klijentskih aplikacija putem slušatelja u stvarnom vremenu te nudi podršku za rad izvan mreže za mobilne i web aplikacije, što omogućuje izradu responzivnih aplikacija koje rade bez obzira na kašnjenja u mreži ili internetsku povezanost. Firestore također omogućuje besprijeckornu integraciju s ostalim Firebase i Google Cloud proizvodima, uključujući Cloud Run funkcije [25].



Sl. 3.2. Primjer kolekcije korisnika u Firestore-u

Firebase Autentifikacija (engl. *Authentication*) pruža jednostavne i sigurne metode za autentifikaciju korisnika. Omogućuje prijavu putem različitih metoda, uključujući *e-mail* i lozinku,

kao i putem popularnih identitetskih pružatelja poput Googlea, Facebooka i Twittera. Ova usluga pojednostavljuje proces prijave i registracije korisnika, smanjujući potrebu za razvojem vlastitih sigurnosnih rješenja. Firebase *Authentication* također osigurava zaštitu korisničkih podataka i podržava više faktorsku autentifikaciju, što dodatno povećava sigurnost.



The screenshot shows the Firebase Authentication console interface. At the top, there is a search bar with the text "Search by email address, phone number, or user UID" and an "Add user" button. Below the search bar is a table with the following columns: Identifier, Providers, Created, Signed In, and User UID. The table contains three rows of user data. At the bottom right, there is a pagination control showing "Rows per page: 50" and "1 - 3 of 3".

Identifier	Providers	Created ↓	Signed In	User UID
programer@programer...	✉	Jun 19, 2024	Jun 21, 2024	dtndWp1NdEXq7FmzsHyprW...
tester@tester.com	✉	Jun 19, 2024	Jun 21, 2024	lwN0pw74k5UnlqS2MFtikjIO...
mando.maznik@gmail...	✉	Apr 19, 2024	Jun 21, 2024	FPWNQ47krMNVzE4NZ1z11T...

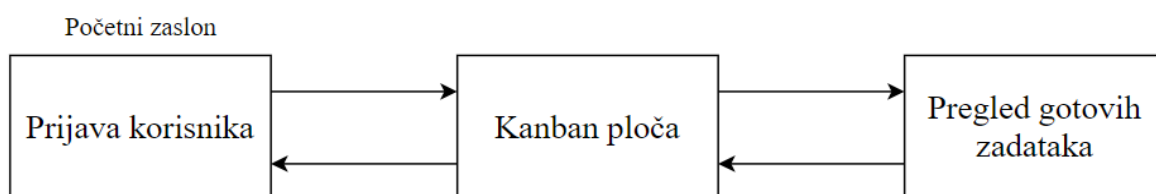
Sl. 3.3. Primjer Firebase *Authentication*

Kombinacija Firestore i Firebase *Authentication* omogućuje jednostavno i učinkovito upravljanje podacima i korisnicima unutar aplikacije, to omogućuje fleksibilno rješenje koje se prilagođava rastu aplikacije. Ove usluge ne samo da poboljšavaju performanse aplikacije, već i osiguravaju sigurnost i pouzdanost podataka, što je ključno za postizanje visokokvalitetnog korisničkog iskustva.

4. RJEŠENJE

Za izradu web aplikacije digitalna Kanban ploča, odlučeno je koristiti React okvir (eng. *framework*), zajedno s Firebase kao online bazom podataka. Visual Studio Code (VS Code) je odabran kao glavno razvojno okruženje zbog svoje fleksibilnosti, podrške za proširenja i alate za otklanjanje pogrešaka.

Aplikacija se sastoji od tri stranice, a to su: stranica za prijavu korisnika, stranica na kojoj se nalazi Kanban ploča, te treća stranica za pregled izvršenih zadataka. Iako aplikacija ima više stranica, one koriste pristup jednostranične aplikacije (eng. *Single Page Application*), što znači da se sve funkcionalnosti odvijaju unutar jedne stranice bez ponovnog učitavanja, odnosno bez potrebe za gumbom osvježi (eng. *refresh*). Na slici 4.1. može se vidjeti struktura aplikacije.



Sl. 4.1. Struktura aplikacije

U ovom poglavlju detaljnije će se pokazati i pojasniti ključne dijelove koda na kojima se bazira aplikacija. Obradit će se način na koji su implementirane glavne funkcionalnosti kao što su dodavanje i upravljanje zadacima na Kanban ploči te način na koji je integriran Firebase za pohranu podataka.

Ako se želi dodati zadatak na našu Kanban ploču, korisnik mora biti prijavljen kao *project manager* jer samo on ima mogućnost dodavanja novih zadataka u stupac Ideja. Definicija uloge *project manager* je korištena slobodno. Na slici 4.1. može se vidjeti da prilikom pritiska gumba + (plus), provjerava je li korisnik *project manager*, te ako je otvara se prozor za dodavanje novog zadatka.

```

99     const handleClick = () => {
100         if(userRole === "project manager"){
101             setShowInput(true);
102         }
103         else{
104             alert("Ovlasti: project manager!")
105             return;
106         }
107     };

```

Sl. 4.2. Provjera prijavljenog korisnika

Na slici 4.2. prikazana je implementacija koda za dodavanje novog zadatka. Prvo se provjerava ima li u stupcu mjesta za novi zadatak, odnosno osigurava se da novi zadatak ne prekoračuje maksimalni broj zadataka u stupcu. Zatim se upisuje novi zadatak u bazu podataka i isti taj zadatak se sprema u lokalno stanje kako bi ga se moglo prikazati na Kanban ploči.

```

109     const handleInputSubmit = async (pickedColor, role) => {
110         if (isLoading) return;
111
112         if (ideja.length < MAX_SQUARES_PER_COLUMN) {
113             const newIdeja = { naslov: naslov, opis: opis, color: pickedColor, role: role };
114             try {
115                 const docRef = await addDoc(collection(db, 'ideja'), newIdeja);
116                 console.log('Dokument je uspješno dodan: ' + docRef.id);
117
118                 // Dodajte novu ideju u lokalno stanje ideja
119                 setIdeja([...ideja, { id: docRef.id, ...newIdeja }]);
120             } catch (error) {
121                 console.error('Greška prilikom dodavanja dokumenta:', error);
122                 console.log('Greška prilikom spremanja ideje u bazu podataka.');
```

Sl. 4.3. Predložak za dodavanje novog zadatka

```

141 const handleDrop = async(event, targetColumn) => {
142   event.preventDefault();
143   const data = JSON.parse(event.dataTransfer.getData("text/plain"));
144   const { column, index } = data;
145
146   const restrictions = {
147     ideja: ["izrada", "test", "integriran", "gotov"],
148     plan: ["ideja", "test", "integriran", "gotov"],
149     izrada: ["ideja", "plan", "integriran", "gotov"],
150     test: ["ideja", "plan", "izrada", "gotov"],
151     integriran: ["ideja", "plan", "izrada", "test"],
152     gotov: ["ideja", "plan", "izrada", "test", "integriran"]
153   };
154
155   if (restrictions[column] && restrictions[column].includes(targetColumn)) {
156     alert(`You can't drag to ${targetColumn.charAt(0).toUpperCase() + targetColumn.slice(1)}`);
157     return;
158   }
159
160   // Provjerite je li broj kvadrata u odredišnom stupcu manji od maksimalnog dopuštenog
161   const numSquaresInTargetColumn =
162     targetColumn === "ideja"
163       ? ideja.length
164       : targetColumn === "plan"
165         ? plan.length
166         : targetColumn === "izrada"
167           ? izrada.length
168           : targetColumn === "test"
169             ? test.length
170             : targetColumn === "integriran"
171               ? integriran.length
172               : gotov.length

```

Sl. 4.4. Predložak funkcije *handleDrop()*

```

173
174 if (numSquaresInTargetColumn < MAX_SQUARES_PER_COLUMN) {
175   // Provjera je li stupac odredišta isti kao i stupac izvora
176   if (column !== targetColumn) {
177
178     switch (column) {
179       case "ideja":
180         const draggedIdeja = ideja[index];
181         if (userRole === "programer" || userRole === "project manager"){
182           try {
183             // Brišemo objekt iz kolekcije "ideja"
184             await deleteDoc(doc(db, "ideja", draggedIdeja.id));
185             console.log("Dokument uspješno obrisano iz kolekcije 'ideja'.");
186           } catch (error) {
187             console.error("Greška prilikom brisanja dokumenta iz kolekcije 'ideja':", error);
188           }
189           setIdeja(ideja.filter((_, i) => i !== index));
190
191           switch (targetColumn) {
192             case "plan":
193               const moveDatePlan = new Date();
194               const newDocRef = doc(collection(db, 'plan'));
195               const newDocId = newDocRef.id;
196               const newDocData = { ...draggedIdeja, moveDatePlan, id: newDocId, programer: userName };
197
198               try {
199                 await setDoc(newDocRef, newDocData); // Dodajte dokument u Firestore s postavljenim ID-om
200                 console.log("Dokument uspješno dodan u kolekciju 'plan' s ID:", newDocId);
201               } catch (error) {
202                 console.error("Greška prilikom dodavanja dokumenta u kolekciju 'plan':", error);
203               }
204               setPlan([...plan, newDocData]);
205               break;
206             default:
207               You, last month + Dodavanje ovlasti korisnicima
208               break;
209           }
210         } else {
211           alert("Ovlasti: programer!")
212           return;
213         }
214       break;

```

Sl. 4.5. Predložak funkcije *handleDrop()*

Glavna funkcija ove aplikacije je *handleDrop()*. Kada korisnik premjesti zadatak iz jednog stupca u drugi, funkcija *handleDrop()* se aktivira. Ova funkcija ne samo da ažurira bazu podataka dodavanjem i brisanjem odgovarajućih zapisa, već također odmah osvježava prikaz na ploči kako bi odražavala promjene u stvarnom vremenu.

Na slici 4.3. može se vidjeti kako se provjerava da se zadatak iz određenog stupca ne prebaci u nedozvoljeni stupac te se odmah korisniku šalje obavijest kako to nije moguće. Također, provjerava se da broj zadataka u svakom stupcu ne premašuje dozvoljeni broj.

Na slici 4.4. može se vidjeti kako se prvo provjerava da stupac ne premašuje dozvoljeni broj zadataka. Ukoliko premašuje, korisnik dobiva obavijest o tome. Zatim, na slici 4.5. može se vidjeti da se provjerava da korisnik nije prebacio zadatak iz određenog stupca u taj isti stupac. Nakon toga postoji *switch-case* struktura grananja kojoj predajemo stupac iz kojeg prebacujemo zadatak. U tom *case*-u se provjerava ima li korisnik ovlasti za pomicanje zadatka iz tog stupca, te se zatim taj zadatak briše iz tog stupca u bazi podataka i lokalno. Nakon toga postoji drugi *switch-case* kojem se predaje stupac u koji se želi prebaciti zadatak. U tom *case*-u se dodaje taj zadatak u bazu podataka tog stupca te se on sprema lokalno kako bi ga mogli prikazati na ploči.

Nadalje, postoje ponovni *switch-case*-ovi za svaki izvorni i odredišni stupac, koji su slični ovim *switch-case*-ovima. Jedina razlika je u tome što koriste različite podatke za stupce i provjere korisnika.

```
1  import {initializeApp} from 'firebase/app';
2  import {getAuth} from 'firebase/auth';
3  import {getFirestore} from "firebase/firestore";
4
5  const firebaseConfig = {
6    apiKey: "AIzaSyDq03NLgaDDPe02YjoXlMeeTn-n_x6dDn4",
7    authDomain: "digitalna-kanban-ploca.firebaseio.com",
8    projectId: "digitalna-kanban-ploca",
9    storageBucket: "digitalna-kanban-ploca.appspot.com",
10   messagingSenderId: "976602619463",
11   appId: "1:976602619463:web:e42923917b5b5b56109b62"
12 };
13
14 const app = initializeApp(firebaseConfig);
15
16 const auth = getAuth();
17
18 const db = getFirestore(app)
19
20 export { app, auth, db };
```

Na slici 4.6. može se vidjeti predložak koji prikazuje konfiguraciju Firebase-a. Taj dio koda služi za povezivanje Firebase-a s aplikacijom te omogućava korištenje njegovih alata, kao što su *Firestore* i *Authentication* u primjeru.

5. UPUTE ZA KORIŠTENJE APLIKACIJE

Aplikacija Digitalna Kanban ploča može imati više korisnika, ali postoje samo tri uloge: *Project Manager*, *Programer* i *Tester*. Svaka uloga ima različite mogućnosti, odnosno ovlasti. Kao što se u prijašnjem poglavlju objasnilo, aplikacija se sastoji od tri stranice. Jedna stranica je za prijavu korisnika, druga stranica je sama Kanban ploča, a treća stranica služi za pregled izvršenih zadataka.

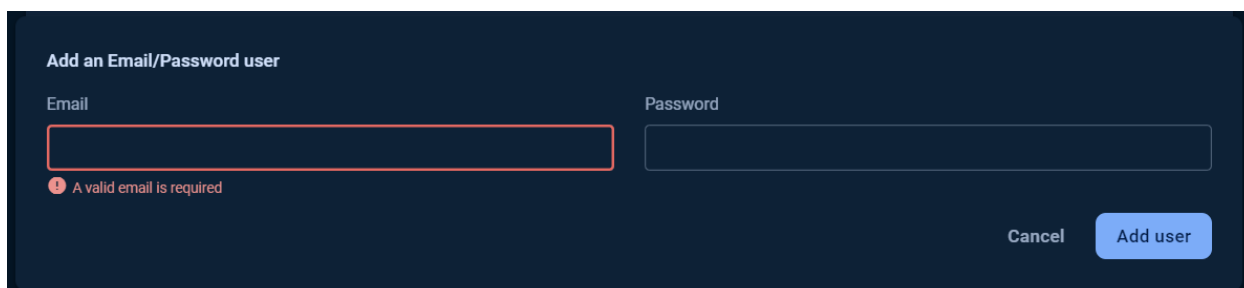
Za početak je potrebno stvoriti korisnike i dodijeliti im uloge. Korisnici se stvaraju izravno u bazi podataka u Firebase *Authentication*-u. Na slici 5.1. prikazani su prethodno dodani korisnici.



Identifier	Providers	Created ↓	Signed In	User UID
programer@programer...	📧	Jun 19, 2024	Jun 21, 2024	dtndWp1NdEXq7FmzsHyprW...
tester@tester.com	📧	Jun 19, 2024	Jul 24, 2024	lwN0pw74k5UnlqS2MFtikjIO...
mando.maznik@gmail...	📧	Apr 19, 2024	Jul 24, 2024	FPWNQ47krMNVzE4NZ1z11T...

Sl. 5.1. Trenutni korisnici

Ako se želi dodati novog korisnika, potrebno je pritisnuti gumb „Add user“. Na slici 5.2. vidi se kako se korisniku dodjeljuje *e-mail* i lozinka, kako bi se mogao prijaviti u aplikaciju. Svakom korisniku automatski se generira jedinstveni identifikacijski broj (eng. *User UID*).



Add an Email/Password user

Email

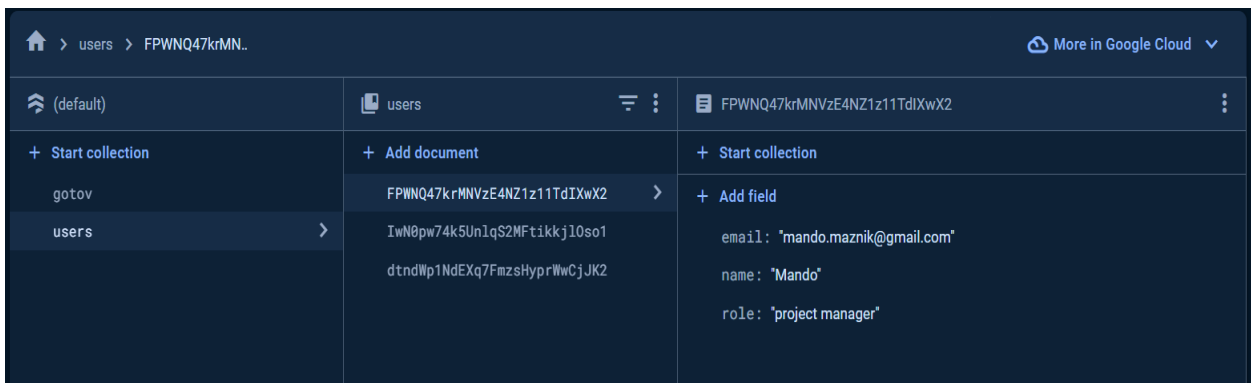
Password

🔴 A valid email is required

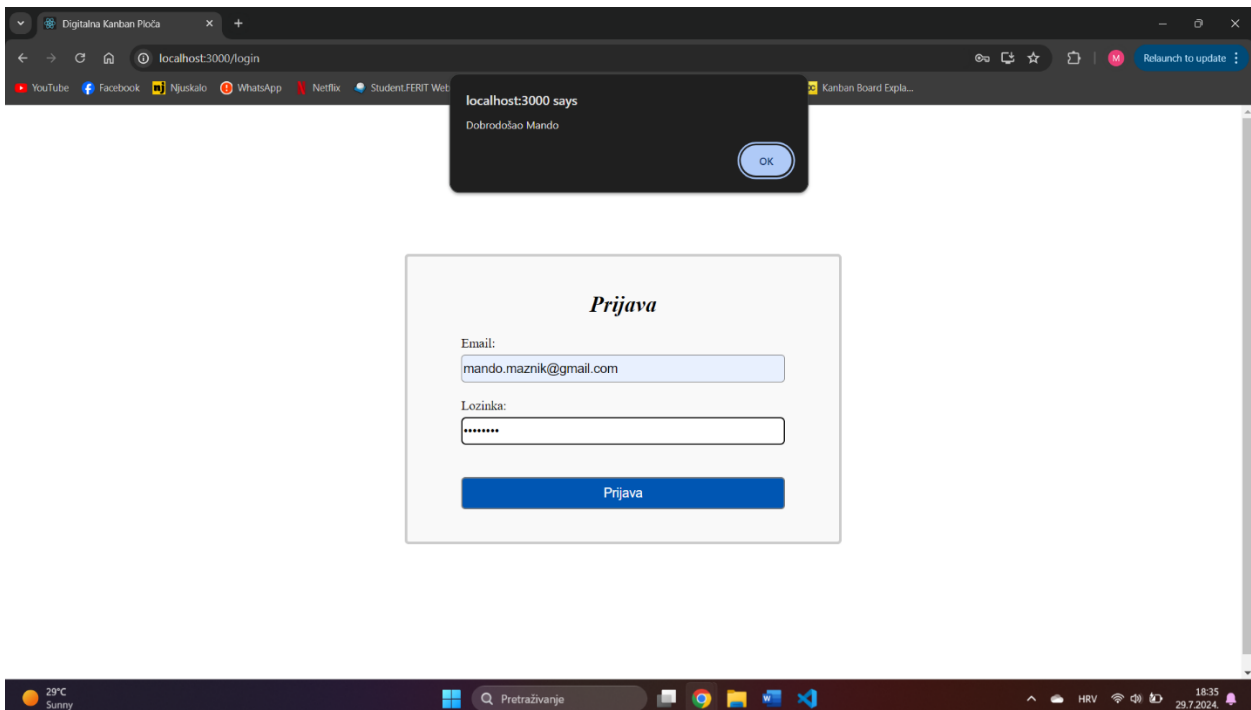
Cancel

Sl. 5.2. Dodavanje novog korisnika

Zatim je potrebno korisnike koji su stvoreni u *Firestore Authentication*-u povezati s *Firestore*-om. Povezuju se putem jedinstvenog identifikacijskog broja korisnika (*User UID*). Na slici 5.3. može se vidjeti kolekcija „users“. Kako bi se dodao novi korisnik, potrebno je pritisnuti gumb „+ Add document“, te se za „document ID“ postavlja jedinstveni identifikacijski broj korisnika (*User UID*) korisnika koji se dodao u *Firestore Authentication*-u. Zatim se dodaju polja: *e-mail*, *name* i *role*, te za svako od tih polja unosimo *e-mail* korisnika, ime i ulogu.

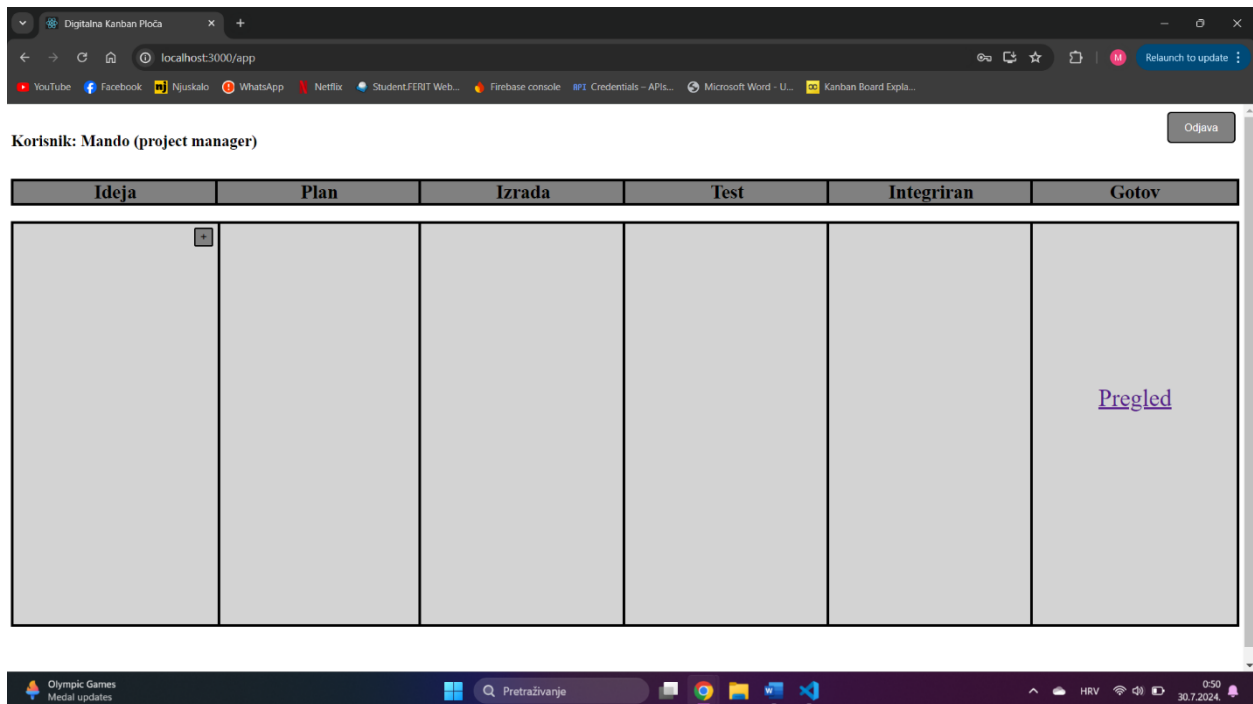


Sl. 5.3. Korisnici u Firestore-u



Sl. 5.4. Stranica za prijavu

U aplikaciji je početna stranica prijava korisnika (Slika 5.4.). Korisnici se prijavljuju s podacima koji su im dodijeljeni u Firebase-u. Nakon uspješne prijave, korisnik dobiva obavijest s porukom „Dobrodošao [ime korisnika]“, kao što se vidi u primjeru gdje piše „Dobrodošao Mando“. Ta obavijest se razlikuje od korisnika do korisnika jer ovisi o imenu korisnika koji se prijavio. Ukoliko *e-mail* ili lozinka nisu ispravni, korisnik će dobiti obavijest "Pogreška prilikom prijave."



Sl. 5.5. Stranica s Kanban pločom

Nakon što se korisnik uspješno prijavi, otvara se stranica s Kanban pločom, koja je ujedno glavna stranica aplikacije. Na slici 5.5. može se vidjeti kako je na toj stranici Kanban ploča. U gornjem lijevom kutu prikazano je ime korisnika koji je trenutno prijavljen te njegova uloga. Također, na toj stranici nalazi se i dugme za odjavu korisnika.

Ukoliko korisnik kojemu je uloga *project manager* pritisne „+“ u polju Ideja, otvara se prozor u kojem se može dodati novi zadatak (Slika 5.6.). Novi zadatak sadrži: naslov, opis te tri dugmeta koja predstavljaju komponente razvoja. Pritiskom na neko dugme stvara se novi zadatak koji će biti u boji ovisno o tome koje dugme se pritisnulo.

The image shows a dialog box for adding a new task. It contains the following elements:

- Naslov:** A text input field with a placeholder text "Maksimalni broj znakova: 37."
- Opis:** A larger text area for describing the task.
- FRONTEND:** A purple button.
- BACKEND:** A blue button.
- DESIGN:** A green button.
- Return:** A grey button.

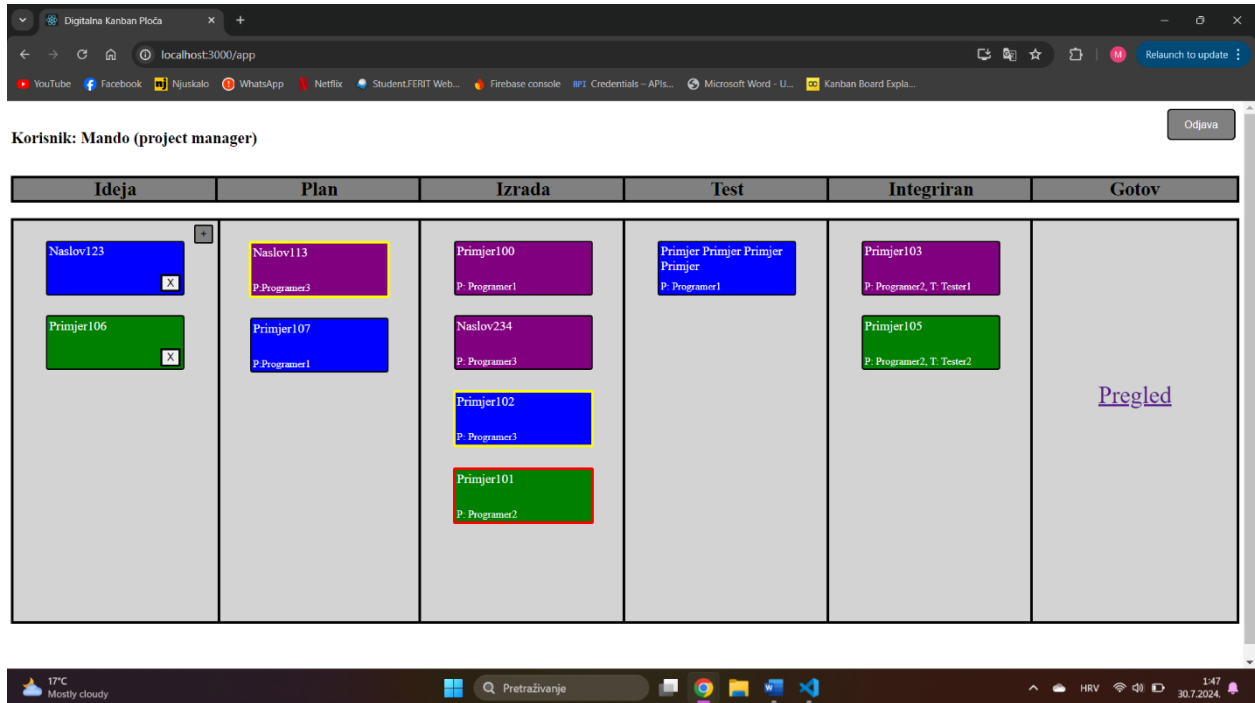
Sl. 5.6. Prozor za dodavanje novog zadatka

Na slici 5.7. može se vidjeti kako ploča izgleda kada su na njoj postavljeni zadaci. Na svakom zadatku piše naslov, slovo „P“ koje označava *Programer* i slovo „T“ koje označava *Tester*. Korisnik ima mogućnost pomicanja zadataka iz polja u polje, ali to ovisi o njegovim ovlastima. *Programer* može pomicati zadatke iz polja Ideja pa sve do polja Test, dok *tester* može pomicati zadatke samo iz polja Test u polje Integriran. *Project manager* zatim pomiče zadatke u polje Gotov.

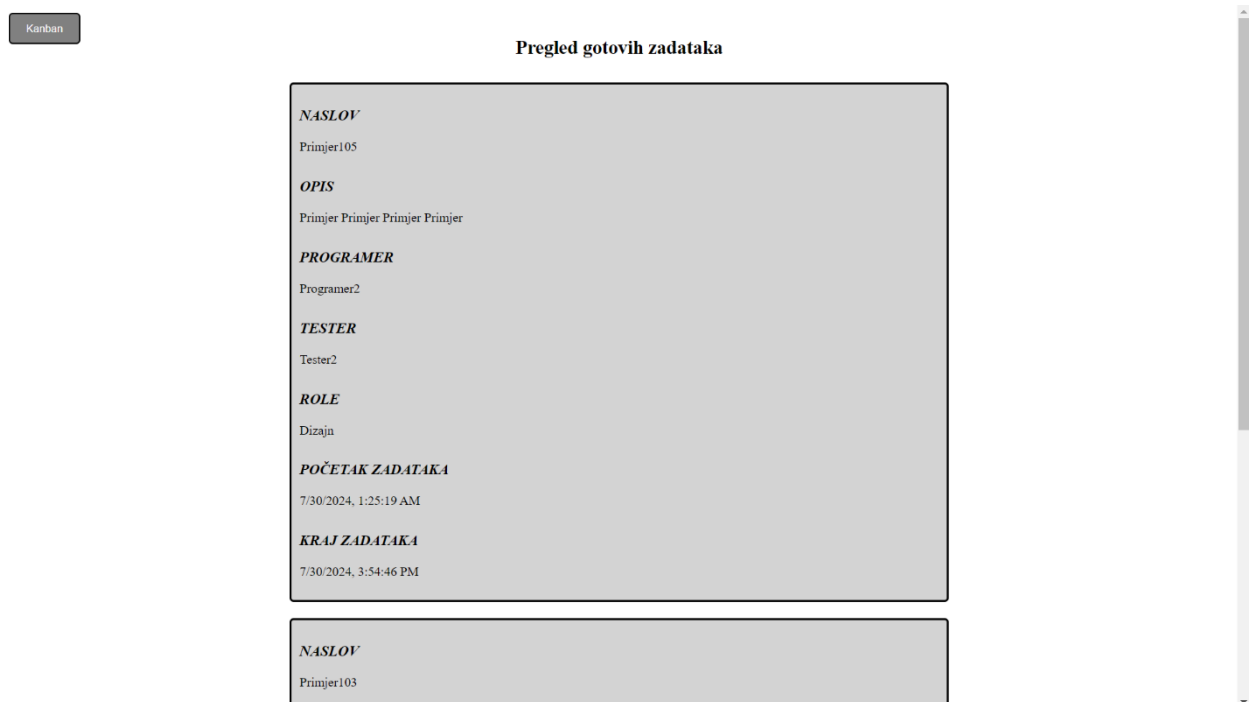
U polju Ideja može se vidjeti dugme „X“ na zadacima. Pritiskom na to dugme briše se zadatak iz polja Ideja. Nadalje, kada korisnik s ulogom *programera* pomakne zadatak iz polja Ideja u polje Plan, tom zadatku se pridjeljuje ime tog *programera*, koje se upisuje u bazu podataka te na taj zadatak. Isto tako, kada korisnik s ulogom *testera* pomakne zadatak iz polja Test u polje Integriran, tom zadatku se pridjeljuje ime tog *testera*, koje se također upisuje u bazu podataka te na taj zadatak.

Svako polje može sadržavati maksimalno 4 zadatka. Za svaki zadatak u svim stupcima postoji vremensko ograničenje i tolerancija na to ograničenje. Ukoliko je zadatak duže od tri dana, a manje od pet dana u tom stupcu, taj zadatak poprimi žuti okvir. Postoji tolerancija od dva dana, odnosno,

ako je zadatak duže od pet dana u tom stupcu, okvir zadatka postaje crvene boje. Ovo služi kako bi pratili da zadatak nije predugo u pojedinim stupcima zbog vremenskih ograničenja.



Sl. 5.7. Kanban ploča s dodanim zadacima



Sl. 5.8. Stranica za pregled gotovih zadataka

U polju Gotov postoji veza (eng. *link*) „Pregled“ koja vodi prema stranici za pregled gotovih zadataka. Nakon što *project manager* prebaci zadatak iz polja Integriran u polje Gotov, zadatak se sprema u bazu podataka kao gotov. Na slici 5.8. mogu se vidjeti gotovi zadatci s detaljnim prikazom informacija kao što su: naslov, opis, *programer*, *tester*, komponenta razvoja (*role*), početak zadatka te kraj zadatka. Uz to, postoji dugme „Kanban“ putem kojeg se može vratiti na stranicu s Kanban pločom. Ova stranica omogućuje bolji uvid u zadatke jer se vidi tko je radio na zadatku, kada je zadatak započet i kada je završen.

Aplikacija je uspješno prošla kroz faze testiranja, uključujući provjere korisničkog sučelja. Svi ključni aspekti aplikacije, kao što su dodavanje, premještanje i brisanje kartica, funkcioniraju prema dogovorenim specifikacijama i očekivanjima. Testiranje je potvrdilo da aplikacija pruža stabilno i pouzdano korisničko iskustvo.

6. ZAKLJUČAK

Cilj ovog rada bio je objasniti koncept Kanban ploče i razviti jednostavnu aplikaciju koja prikazuje njezinu funkcionalnost. Kroz ovaj projekt koristi se React, popularni JavaScript okvir za izgradnju korisničkih sučelja, te Firebase, platformu za razvoj mobilnih i web aplikacija. Razvoj aplikacije odvijao se u Visual Studio Code, prilagodljivom razvojnom okruženju.

Kanban ploča pokazala se kao izuzetno koristan vizualni alat za upravljanje zadacima, omogućujući timovima da prate napredak rada, identificiraju zastoje i poboljšaju ukupnu učinkovitost. U našoj aplikaciji, korisnici mogu dodavati, premještati i pregledavati zadatke unutar različitih stupaca koji predstavljaju faze razvoja, kao što su ideja, plan, izrada, test, integriran i završetak.

React je omogućio modularan i responzivan dizajn korisničkog sučelja, što je značajno olakšalo razvoj i održavanje aplikacije. Korištenjem Reacta, aplikacija je podijeljena na manje, ponovno upotrebljive komponente, što je omogućilo učinkovitiji razvoj i lakše upravljanje kodom. Ova komponentna arhitektura omogućila je jednostavnu implementaciju složenih funkcionalnosti te brze promjene i prilagodbe korisničkog sučelja. Firebase je pružio snažan backend, uključujući Firestore za pohranu podataka u stvarnom vremenu i Firebase *Authentication* za sigurnu prijavu korisnika.

Kombinacijom ovih tehnologija postigli smo skalabilno i sigurno rješenje koje korisnicima omogućava pregled zadataka i njihovih statusa tijekom razvoja. Ovaj rad ne samo da prikazuje tehničke aspekte izrade digitalne Kanban ploče, već također naglašava praktične prednosti koje ovakav sustav može donijeti u upravljanju projektima. Uz to, demonstrira kako se moderne tehnologije mogu učinkovito koristiti za poboljšanje organizacije i produktivnosti unutar timova.

LITERATURA

- [1] Kanban Tool, History of Kanban [online], Kanban Tool, Katowice, Poljska, dostupno na: <https://kanbantool.com/kanban-guide/kanban-history> [06.05.2024.]
- [2] Izvorna slika, dostupno na: <https://kanbanzone.com/resources/kanban/> [06.05.2024.]
- [3] Izvorna slika, dostupno na: <https://blog.limes.hr/kanban-metoda-fleksibilna-metoda-za-bolju-produktivnost/> [19.05.2024.]
- [4] Kanban Tool, Kanban board [online], Kanban Tool, Katowice, Poljska, dostupno na: <https://kanbantool.com/kanban-board> [19.05.2024.]
- [5] Andrew Stellman, Jennifer Greene, Learning Agile Understanding Scrum, XP, Lean, and Kanban, O'Reilly Media, 2014.
- [6] Leankor Kanban for dummies, Leankor.com, 2015.
- [7] Trello, About Trello [online], Atlassian, Sydney, Australija, dostupno na: <https://trello.com/templates/engineering/kanban-dev-board-lvRpONoJ> [12.05.2024.]
- [8] Izvorna slika, dostupno na: <https://planyway.com/blog/trello-kanban-principles> [12.05.2024.]
- [9] Jira, About Jira [online], Zapier, San Francisco, Sjedinjene Američke Države, dostupno na: <https://zapier.com/blog/trello-vs-jira/> [14.05.2024.]
- [10] Izvorna slika, dostupno na: <https://deviniti.com/blog/application-lifecycle-management/jira-ticket-epic-templates-ultimate-guide/> [14.05.2024.]
- [11] Asana, About Asana [online], Process Street, San Francisco, Sjedinjene Američke Države, dostupno na: <https://www.process.st/how-to/asana-kanban/> [19.05.2024.]
- [12] Izvorna slika, dostupno na: <https://blog.asana.com/2016/11/introducing-boards/> [19.05.2024.]
- [13] Izvorna slika, dostupno na: <https://www.pm-partners.com.au/insights/the-agile-journey-a-scrum-overview/> [21.05.2024.]
- [14] Kanban Training Book, International Scrum Institute, 2020
- [15] Izvorna slika, dostupno na: <https://www.planview.com/resources/guide/lean-principles-101/what-is-lean/> [12.09.2024.]

- [16] Lean, About Lean [online], Javatpoint, Noida, India, dostupno na: <https://www.javatpoint.com/lean-vs-kanban> [22.05.2024.]
- [17] Kanban in Manufacturing, Kanban Inventory Management [online], Austin, Sjedinjene Američke Države, dostupno na: <https://www.projectmanager.com/blog/kanban-in-manufacturing> [22.05.2024.]
- [18] Izvorna slika, dostupno na: <https://agilefixer.com/2017/02/07/scrum-kanban-importance-of-visualisation/> [22.05.2024.]
- [19] How to implement Kanban in healthcare Management, Kanban in healthcare [online], teamhub, London, Engleska, dostupno na: <https://teamhub.com/blog/how-to-implement-kanban-in-healthcare-management/> [23.05.2024.]
- [20] Kanban boards for Agile education, Kanban in education [online], Medium, San Francisco, Sjedinjene Američke Države, dostupno na: https://medium.com/@jessica_15943/kanban-boards-for-agile-education-enhancing-teacher-parent-relationships-d2ae0ed14d6c, [25.05.2024.]
- [21] Izvorna slika, dostupno na: <https://tododoingdone.com/2017/10/kanban-ploca-za-peti-razred/> [25.05.2024.]
- [22] The Best Guide to Know What Is React, About React [online], Simplilearn, San Francisco, Sjedinjene Američke Države, dostupno na: <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs> [16.07.2024.]
- [23] What is Dom in React, About Dom [online], Javatpoint, Noida, India, dostupno na: <https://www.javatpoint.com/what-is-dom-in-react> [12.09.2024.]
- [24] Firebase - introduction, About Firebase [online], GeeksforGeeks, Noida, India, dostupno na: <https://www.geeksforgeeks.org/firebase-introduction/> [18.07.2024.]
- [25] Firestore overview, About Firestore [online], Google, California, Sjedinjene Američke Države, dostupno na: <https://cloud.google.com/firestore/docs/overview> [12.09.2024.]

SAŽETAK

U ovom radu objašnjena je Kanban ploča kao alat za vizualno upravljanje zadacima i procesima. Rad se fokusira na razvoj digitalne verzije Kanban ploče, koja omogućava korisnicima učinkovito praćenje i organizaciju zadataka u timskom radu. Aplikacija pruža funkcionalnosti kao što su dodavanje, premještanje zadataka i pregled gotovih zadataka, čime se poboljšava preglednost i produktivnost. Razvoj digitalne Kanban ploče demonstrira kako tehnologija može unaprijediti tradicionalne metode upravljanja projektima, prilagođavajući ih modernim potrebama korisnika.

Ključne riječi: Kanban, organizacija, produktivnost, upravljanje, vizualizacija

SUMMARY

Digital Kanban board

In this paper, the Kanban board is explained as a tool for visual management of tasks and processes. The paper focuses on the development of a digital version of the Kanban board, which allows users to efficiently track and organize tasks in teamwork. The application provides functionalities such as adding, moving tasks, and reviewing completed tasks, whereby improving visibility and productivity. The development of the digital Kanban board demonstrates how technology can enhance traditional project management methods, adapting them to the modern needs of users.

Keywords: Kanban, management, organization, productivity, visualization.