



УНИВЕРЗИТЕТ У НОВОМ САДУ ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
НОВИ САД

Департман за рачунарство и аутоматику

Одсек за рачунарску технику и рачунарске комуникације

ЗАВРШНИ (BACHELOR) РАД

Кандидат: Јован Зубовић

Број индекса: РА5/2019

Тема рада: Даљинска конфигурација боја на Google TV преко Firebase сервера

Ментор рада: проф. Др Илија Башичевић

Нови Сад, септембар, 2023



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР :			
Идентификациони број, ИБР :			
Тип документације, ТД :		Монографска документација	
Тип записа, ТЗ :		Текстуални штампани материјал	
Врста рада, ВР :		Завршни (Bachelor) рад	
Аутор, АУ :		Јован Зубовић	
Ментор, МН :		Проф. Др Илија Башичевић	
Наслов рада, НР :		Даљинска конфигурација боја на Google TV преко Firebase сервера	
Језик публикације, ЈП :		Српски / ћирилица	
Језик извода, ЈИ :		Српски	
Земља публикавања, ЗП :		Република Србија	
Уже географско подручје, УГП :		Војводина	
Година, ГО :		2023	
Издавач, ИЗ :		Ауторски репринт	
Место и адреса, МА :		Нови Сад; трг Доситеја Обрадовића 6	
Физички опис рада, ФО : (поглавља/страна/ цитата/табела/слика/графика/прилога)			
Научна област, НО :		Електротехника и рачунарство	
Научна дисциплина, НД :		Рачунарска техника	
Предметна одредница/Кључне речи, ПО :			
УДК			
Чува се, ЧУ :		У библиотеци Факултета техничких наука, Нови Сад	
Важна напомена, ВН :			
Извод, ИЗ :		У последњих неколико година је дошло до великог повећања количине мултимедијалног садржаја доступног на тржишту. То се десило и са паметним телевизорима, где се разни произвођачи међусобно такмиче ко ће понудити најбоље и најзанимљивије искуство својим корисницима. Једна од ставки које су најприметније и које највише утичу на наше искуство су боје приказане на уређају. У раду је реализована конфигурација боја на Google TV преко Firebase конзоле. Коришћен је комерцијални сет-топ-бокс и Гуглова апликација за паметни ТВ.	
Датум прихватања теме, ДП :			
Датум одбране, ДО :			
Чланови комисије, КО :	Председник:	Проф. Др Дарко Чапко	
	Члан:	Проф. Др Жељко Лукач	Потпис ментора
	Члан, ментор:	Проф. Др Илија Башичевић	

	UNIVERSITY OF NOVI SAD • FACULTY OF TECHNICAL SCIENCES 21000 NOVI SAD, Trg Dositeja Obradovića 6	
	KEY WORDS DOCUMENTATION	

Accession number, ANO :		
Identification number, INO :		
Document type, DT :		Monographic publication
Type of record, TR :		Textual printed material
Contents code, CC :		Bachelor Thesis
Author, AU :		Jovan Zubović
Mentor, MN :		Prof. Dr Ilija Bašičević
Title, TI :		Remote configuration of colors on Google TV using Firebase server
Language of text, LT :		Serbian
Language of abstract, LA :		Serbian
Country of publication, CP :		Republic of Serbia
Locality of publication, LP :		Vojvodina
Publication year, PY :		2023
Publisher, PB :		Author's reprint
Publication place, PP :		Novi Sad, Dositeja Obradovica sq. 6
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)		
Scientific field, SF :		Electrical Engineering
Scientific discipline, SD :		Computer Engineering, Engineering of Computer Based Systems
Subject/Key words, S/KW :		
UC		
Holding data, HD :		The Library of Faculty of Technical Sciences, Novi Sad, Serbia
Note, N :		
Abstract, AB :		In recent years there has been a large increase in amount of multimedia content available on market. That is also the case with smart television, where various manufacturers compete with each other in who will offer best and most interesting experience to their users. One of the most noticeable components for our experience are the colors shown on the device. In this solution configuration of colors on Google TV is realized with Firebase console. Device used is commercial set-top-box along with Google smart TV application.
Accepted by the Scientific Board on, ASB :		
Defended on, DE :		
Defended Board, DB :	President:	Prof. Dr Darko Čapko
	Member:	Prof. Dr Željko Lukač
	Member, Mentor:	Prof. Dr Ilija Bašičević
		Menthor's sign

Захвалност

Захваљујем се институту РТ-РК на пруженој могућности за реализацију овог рада, као и ментору проф. др Илији Башичевићу на пруженој помоћи током израде завршног (бечелор) рада.

Посебно желим да се захвалим техничком ментору Дејану Нађу на вођењу, стручним и личним саветима током израде завршног рада.

Такође желим да се захвалим породици и пријатељима на пруженој подршци током досадашњег школовања.



УНИВЕРЗИТЕТ У НОВОМ САДУ

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



Садржај

1.	Увод	1
2.	Теоријске основе.....	2
2.1	Телевизија.....	2
2.2	Паметна телевизија.....	3
2.3	HTML веб странице	4
2.4	Firebase	4
2.5	Андроид оперативни систем.....	5
2.6	Андроид Студио.....	7
2.7	Хексадецимални приказ боја	8
3.	Концепт решења	9
3.1	Коришћени алати и платформе	9
3.2	Опис решења	9
4.	Програмско решење	11
4.1	Програмско решење у HTML	11
4.1.1	Изглед странице.....	11
4.1.2	Инсталирање Firebase и подешавање пројекта.....	11
4.1.3	Функционалност веб странице у Јаваскрипту	16
4.2	Програмско решење у Андроид Студију.....	19
5.	Тестирање и верификација	21
6.	Могућа унапређења	26
7.	Закључак.....	27
8.	Литература.....	28

Списак слика

- Слика 1 - Google TV на ком су промењене боје приказа
- Слика 2 - Старински телевизор из 60-тих година
- Слика 3 – Дигитални телевизор новије генерације
- Слика 4 - Паметан телевизор
- Слика 5 - Изглед Веб странице прављене у HTML
- Слика 6 - Изглед базе у реалном времену на Firebase платформи
- Слика 7 - Лого Андроида
- Слика 8 - Софтверски стек Андроида
- Слика 9 - Изглед Андроид Студио окружења, са претходно наведеним целинама видљивим са леве стране, и неколико емулятора са десне стране
- Слика 10 - Грецки кутија са даљинским управљачем
- Слика 11 - Инсталирање Node.js
- Слика 12 - Провера успешне инсталације
- Слика 13 - Први корак у инсталацији Firebase
- Слика 14 - Други корак у инстаалцији Firebase
- Слика 15 - Креирање новог пројекта
- Слика 16 - Први корак у креирању пројекта
- Слика 17 - Опционо дозвољавање Google Analytics
- Слика 18 - Одабир подразумеваног (свог) налога
- Слика 19 - Прављење нове базе у реалном времену
- Слика 20 - Одабир сервера на ком ће база бити сачувана
- Слика 21 – Одабир сигурносних правила()
- Слика 22 - Отварање подешавања пројекта у Firebase
- Слика 23 - Додавање нове веб апликације у пројекат
- Слика 24 - Први корак у додавању базе
- Слика 25 - Други корак у додавању базе
- Слика 26 - Додавање Firebase у Андроид Студио
- Слика 27 - Последњи корак у повезивању Firebase са Андроид Студијом
- Слика 28 - Тест без одабраног шаблона
- Слика 29 - Унос мање од 6 карактера
- Слика 30 - Унос карактера који није валидна хексадецимална цифра
- Слика 31 - Генерисање излазног фајла помоћу кључа
- Слика 32 - Одабир екстензије излазног фајла
- Слика 33 - Попуњавање података о кључу
- Слика 34 - Одабир фолдера где ће бити смештен излазни фајл
- Слика 35 - Преименовани излазни фајл
- Слика 36 - Инсталирање на сет-топ-бокс ради тестирања читања базе у Андроид Студију
- Слика 37 - Приказ протеклог времена до примене промена у милисекундама

Списак табела

- Табела 1 - Тестни случајеви

Скраћенице

HTML - Hyper Text Markup Language, описни језик за опис изгледа веб страница

CSS - Cascading style sheet, језик форматирања за побољшање изгледа HTML странице

RGB - Red Green Blue, чест начин дефинисања боја на електронским уређајима

OS - Оперативни систем

DVB - Digital Video Broadcasting, један од стандарда дигиталне телевизије

ATSC - Advanced Television System Committee, један од стандарда дигиталне телевизије

ISDB - Integrated Services Digital Broadcasting, један од стандарда дигиталне телевизије

DTMB - Digital Terrestrial Multimedia Broadcast, један од стандарда дигиталне телевизије

TV - Телевизор

XML - Extensible markup language, описни језик коришћен у Андроид Студију

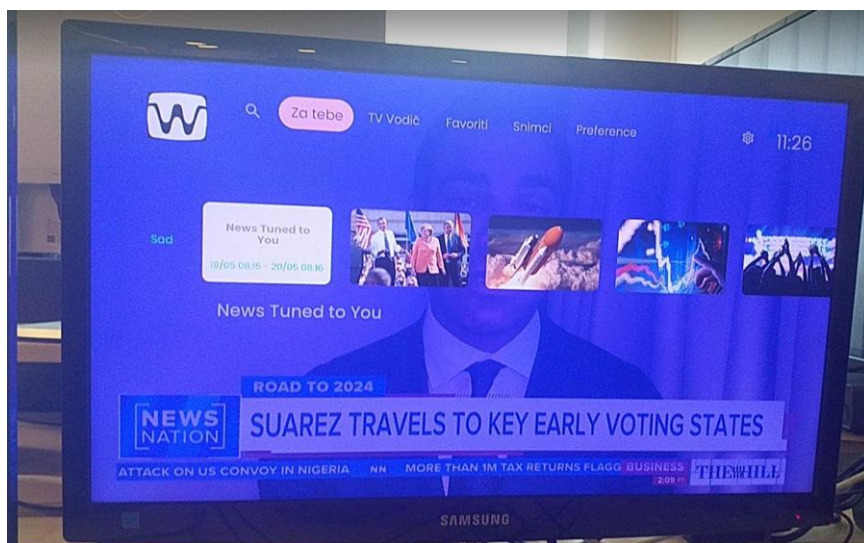
JSON - JavaScript Object Notation, формат за чување података

1. Увод

У iWedia групи компаније РТ-РК доо се већ дуже време развија Google Reference Plus апликација за паметне телевизорe. Једна од многих опција ове апликације је омогућавање мењања боја менија администратору мреже. То се до сада радило преко локалне базе података.

Задатак овог БСц рада је омогућавање оваквог управљања бојама преко Firebase сервера, и то у реалном времену.

У наредним поглављима ће детаљно бити описане теоријске основе, приступ решавању проблема, као и могућа будућа унапређења.



Слика 1 - Google TV на ком су промењене боје приказа

2. Теоријске основе

2.1 Телевизија

Телевизор је уређај за приказ радиодифузних аналогних или дигиталних сигнала преко земаљског, кабловског или сателитског преноса. Преко екрана приказује покретне слике (које се смењују 25-30 пута у секунди), а преко звучника пушта звук.

Телевизијски сигнал се производи у камери, претварањем покретне сцене и звука у електричне сигнале, уз помоћ фотоосетљивог уређаја и микрофона. Ови сигнали се затим емитују преко телевизијских предајника до телевизијске антене повезане са пријемником. У антени се врши раздвајање сигнала за осветљење, боју и звук и њихово слање до телевизора. У телевизору се покретне слике добијају емитовањем слика које се састоје од тачкица у боји.

Телевизори су настали 20-тих година, а у масовну употребу су ушли 50-тих година прошлог века. Тада су се користили аналогни сигнали, а око 2000. се прешло на дигиталне, али и на пренос сигнала преко интернета. Дигитална телевизија је омогућила вишу дефиницију слике, шири однос ширине и дужине екрана и економичније преношење сигнала.

Стандарди дигиталне телевизије су различити, а најчешћи су коришћени следећи:

1. Digital Video Broadcasting (DVB) – Користи се углавном у земљама Европе, Азије, Африке и Океаније
2. Advanced Television System Committee (ATSC) – Користи се у 9 земаља, 8 у Северној Америци и у Јужној Кореји
3. Integrated Services Digital Broadcasting (ISDB) – Користи се у Јапану, Филипинима и већини јужноамеричких држава
4. Digital Terrestrial Multimedia Broadcast (DTMB) – Користи се у Кини, Пакистану, Лаосу у Камбоџи



Слика 2 - Старински ТВ из 60-тих година



Слика 3 - Дигитални ТВ новије генерације

2.2 Паметна телевизија

Паметан или смарт телевизор, понекад хибридни телевизор је спој традиционалне телевизије са приступом интернету и Веб 2.0 карактеристикама и другим интерактивним мултимедијалним садржајима као на пример гледање слика, пуштање музике и слично.

Паметне телевизоре не треба мешати са интернет телевизорима, који су дефинисани чињеницом да садржај примају само преко интернета, док паметни ТВ може да приказује и кабловске и сателитске програме.

Оперативни систем је фабрички инсталиран или доступан преко сет-топ-бокса, а апликације се могу инсталирати преко Google Play-а или App Store-а, као код паметних телефона.

Паметни телевизоры омогућавају повезивање са конзолама за игре, Блу-реј уређајима и осталим мрежним уређајима.



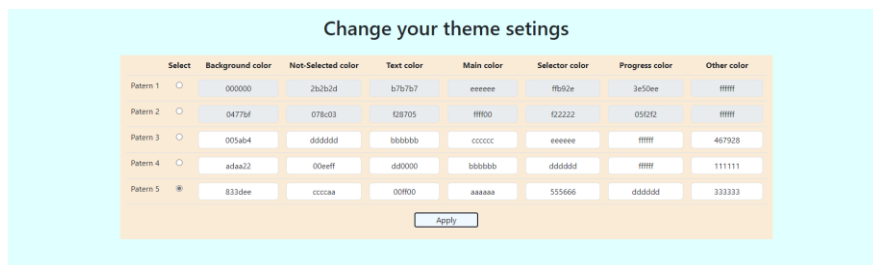
Слика 4 - Паметан телевизор

2.3 HTML веб странице

Веб страница се састоји од елемената и текста. За израду се могу користити разни описни језици, али најчешћи је HTML. Елементи странице се означавају таговима или ознакама.

Тагови могу раздвајати делове странице (<head>, <body>, <script>), елементе унутар тих делова (<title>, <h1>, <table>), описне карактеристике дела текста (, <i>) и могу бити затварајући (пар <title></title>) или незатварајући (
). Све што се стави између отвореног и затвореног тага представљаће текст који ће бити приказан. Функционалност елемената се реализује преко јаваскрипт (енг. javascript) програмског језика.

Коришћењем CSS-а апликацији се може променити изглед по сопственим жељама и потребама. Ово подразумева уређивање боја, величине и фонта текста, ширине, висине елемената и много других параметара који дају програмеру пуну слободу у одабиру изгледа своје странице. CSS може бити написан у посебном документу или се жељени атрибути могу уписати унутар отварајућег тага у оригиналном документу (<input type="submit" id="applyBtn" value="Apply" style="background-color: aliceblue; align-self: center;" width="2em">).

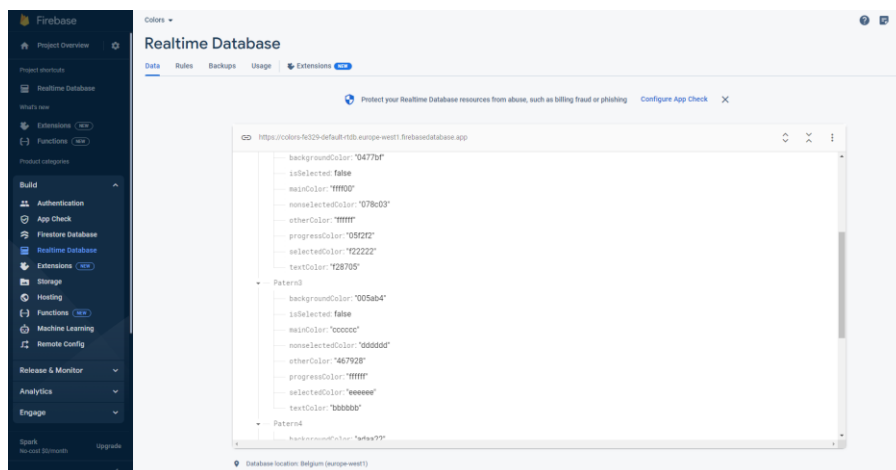


Слика 5 - Изглед Веб странице прављене у HTML

2.4 Firebase

Firebase је скуп бекенд услуга и услуга на облаку које је развио Гугл. Firebase-у се приступа једноставно и бесплатно, преко свог Гугл налога.

Могуће је, између осталог, прављење база у реалном времену, трајних складишта, постављање веб страница, даљинских конфигурација као и интеграција пројеката са Андроид Студијом, HTML веб страницама и iOS пројектима, што чини Firebase једном од најмоћнијих и најпопуларнијих платформи са веома широком заједницом и великом количином бесплатно доступне документације и видео упутстава, како са званичног сајта, тако и од појединачних корисника.



Слика 6 - Изглед базе у реалном времену на Firebase платформи

Firebase чува податке у JSON формату као пар кључ-вредност, на пример Име:Јован, где ``Име`` представља кључ, а ``Јован`` вредност. Овај формат је веома интуитиван и разумљив и за човека и машину, и зато је идеалан када чувамо мању количину података коју треба брзо и лако обрађивати. То је управо случај у овом решењу, где је потребно сачувати хексадецималну вредност неке боје са кључем који јој одговара (боја позадине, боја текста...) и где треба омогућити што бржи упис и читање података како би промена била видљива у реалном времену.

Велика предност Firebase је и у томе што је приступ бази могућ преко сваког уређаја на ком корисник може да приступи свом Гугл налогу, што су данас практично сви уређаји са приступом интернету. Подаци се редовно синхронизују и постоји гаранција да ће на сваком уређају садржај базе бити исти, а свака промена одмах видљива.

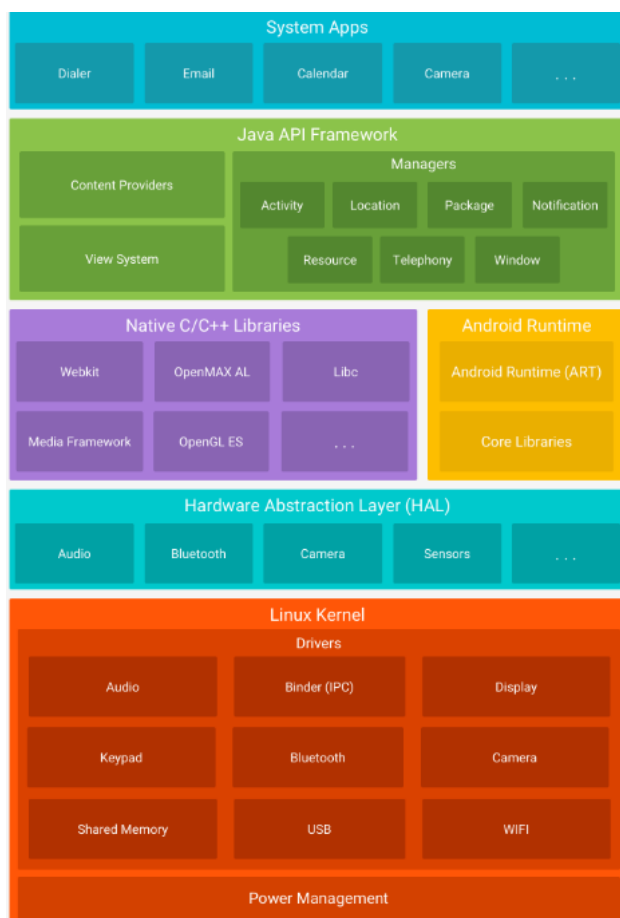
2.5 Андроид оперативни систем

Андроид је оперативни систем настао 2007. (први уређај са овим оперативним системом је пуштен у продају 2008.) као модификација Линукса првенствено намењена за уређаје са екраном на додир. Оперативни систем има отворен изворни код, што значи да свако може да допринесе променама, међутим за велику већину промена је одговоран Гугл. Већина уређаја долази са већ инсталираним Андроидом и унапред инсталираним основним апликацијама као што су Гуглови Chrome претраживач и Play Store за скидање апликација на уређај.

Данас преко 70% паметних телефона користи Андроид, а у међувремену су настале и верзије за паметне телевизоре, конзоле за игре, дигиталне камере и разне друге електронске уређаје.



Слика 7 - Лого Андроида



Слика 8 - Софтверски стек Андроида

На слици изнад се виде слојеви Андроид ОС. Функције слојева који се виде су следеће:

1. Апликативни слој - овде се налазе све апликације које се извршавају. Апликације могу бити системске, корисничке или уграђене
2. Framework (радни оквир) - премошћује разлике између апликативног и нативног слоја, и диктира како се апликације развијају и понашају, и са којим ограничењима.

3. Нативни слој - Садржи велики број библиотека потребних за развој апликација, као и окружење за извршавање апликација
4. Слој хардверске апстракције - Раздваја Андроид од хардвера и омогућава преносивост ОС. Такође дефинише спрегу коју драјвери морају да користе да би радили у Андроиду
5. Линукс слој – Рукује хардверским ресурсима као што су меморија, покренути процеси, приступ камери, звучнику и слично. Линукс се користи као језгро Андроид ОС због своје прилагодљивости и робусности.

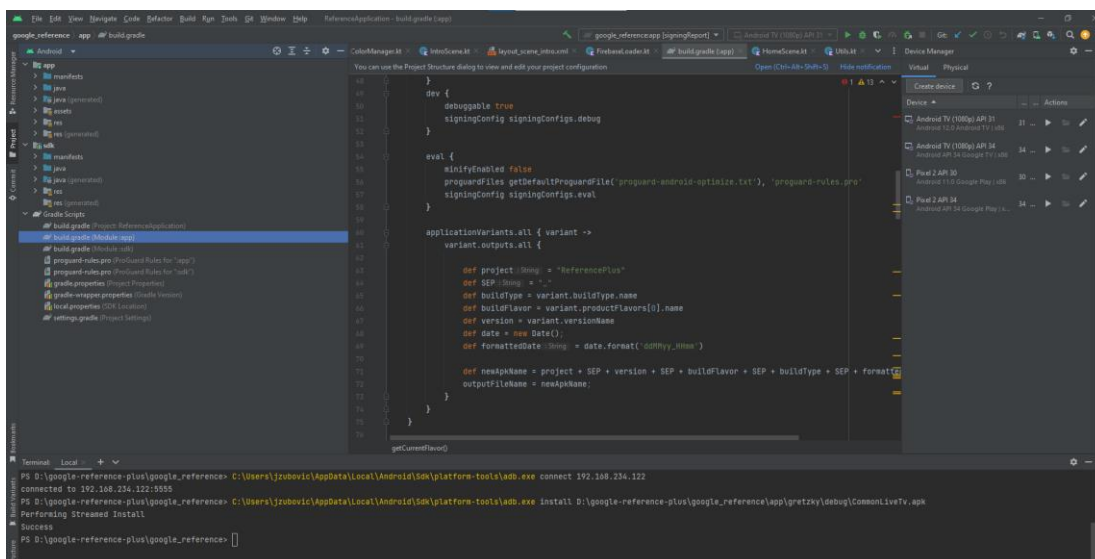
2.6 Андроид Студио

Андроид студио је интегрисано развојно окружење за развој апликација које ће бити инсталиране на Андроид оперативни систем. Ту се подразумевају паметни телефони, телевизори, таблети и други уређаји који подржавају Андроид ОС. Језици који се користе се Јава и Котлин, с тим што је у последње врем Котлин потиснуо Јаву, па је зато коришћен и у овом решењу. Може се користити Ц, али у веома ретким случајевима када је потребна оптимизација веома високог степена.

Пројекат у Андроид Студију се састоји из следећих целина:

1. Манифест - садржи фајл са основним подацима о пројекту и дозволама које користи
2. Јава - Садржи изворни код у Јава или Котлин програмским језицима
3. Ресурси - Садржи описе компонената у неком описном језику (углавном XML), слике, боје, текстове и слично
4. Gradle скрипте – Скрипте за превођење кода и генерисање извршних фајлова

Тестирање је могуће преко виртуелног или стварног уређаја. Виртуелни уређаји се називају и емулятори, и на њима је могуће тестирати апликацију исто као на физичком уређају.



Слика 9 - Изглед Андроид Студио окружења, са претходно наведеним целинама видљивим са леве стране, и неколико емулятора са десне стране

Андроид Студио такође садржи разне корисне помоћне функционалности, као на пример предлоге за аутоматско попуњавање, или аутоматско исправљање неких грешака.

2.7 Хексадецимални приказ боја

Боје се на електронским уређајима често приказују у RGB (Red, Green, Blue) формату, где се црвена, зелена и плава боја мешају у одређеном интензитету и дају остале боје. Интензитет сваке боје се одређује на скали 0-255, или са по две хексадецималне цифре за сваку боју. Хексадецимални систем за основу има број 16, а не 10 као декадни, и са две цифре се покрива потребан опсег (што такође значи да постоји 16 цифара, а не 10). Хексадецималне цифре су: 0-9 као у декадном систему, а цифре које представљају бројеве 10-15 су представљене на следећи начин: A=10, B=11, C=12, D=13, E=14, F=15. Зато је у решењу кориснички унос ограничен на 6 цифара уз проверу да ли је унос валидан у хексадецималном систему.

3. Концепт решења

3.1 Коришћени алати и платформе

За израду рада коришћени су Visual Studio Code и Андроид Студио.

За тестирање на стварном Google TV уређају коришћен је amlogic s905x4 2gb/16gb сет-топ-бокс, такође познат као Грецки.



Слика 10 - Грецки сет-топ-бокс са даљинским управљачем

3.2 Опис решења

Прво је у Visual Studio Code направљен изглед апликације који се може видети на слици 5, и то тако што је коришћен готов Бутстрепов CSS (Бутстреп, енг. Bootstrap, је популарни алат са многим готовим дизајн елементима), а у телу је направљена табела са радио дугметом за одабир жељеног шаблона и пољима за кориснички унос, као и дугме за потврђивање унетих вредности. Интегрисана је и заштита која осигурава да је унос валидан хексидецималан број од 6 цифара.

Пре даљих корака потребно је инсталирати Node.js на свој рачунар и преко њега инсталирати Firebase. Онда треба направити нови Firebase пројекат на свом Гугл налогу и повезати своју апликацију са њим. Тада се може имплементирати функционалност.

У јаваскрипт делу је иницијализована база, додељене су референце на поља Firebase базе и радио дугмићи су повезани са променљивама за одабир шаблона.

Даље се ослушкује промена вредности на референцама која се одмах уписује и на веб страницу, што осигурава да су садржај базе и испис на страници увек исти.

Додат је прислушкивач на клик дугмета за потврду на ком се проверава који шаблон је одабран и на основу тога се одабране вредности уписују на Firebase.

По спецификацији задатка је наведено да се први и други шаблон не мењају.

Након тога је у Андроид студију отворен изворни код Google TV апликације и у њој класа за руковање бојама. У ту класу је затим додата функција која у реалном времену ишчитава садржај базе са Firebasea и прочитане вредности уписује у променљиве које одговарају бојама видљивим на екрану. Ова функционалност је реализована истим корацима као и у HTML-у: иницијализација базе, постављање референци на поља базе, додавање прислушкивача на промену и додела нових вредности у променљиве које одговарају бојама када до промене дође.

4. Програмско решење

4.1 Програмско решење у HTML

4.1.1 Изглед странице

Пратећи опис решења, прво у пројекат укључити Бутстрепов CSS. Он долази са готовим и визуелно лепим атрибутима.

Додавање Бутстрепа:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJ0Z" crossorigin="anonymous">
```

Сада можемо почети са прављењем табеле. Редом с лева на десно треба да имамо текст са редним бројем шаблона, радио дугме помоћу којег ће он бити одабран и онда поља за кориснички унос. Оваквих редова треба да буде 5.

Пример дефинисања реда који се састоји од текста, једног радио дугмета и једног поља за кориснички унос, са потребним атрибутима:

```
<tbody>
<tr><td style="width: 6%;">Patern 1</td>
<td style="width: 2%;"><input style="height: max-content; width: max-content;"
type="radio" name="selected" value="r1" id="r1" required></td>
<td><input type="text" id="1-1" class="form-control" maxlength="6" disabled></td>
...</tr></tbody>
```

Испод табеле додати и дугме за потврду уноса на следећи начин:

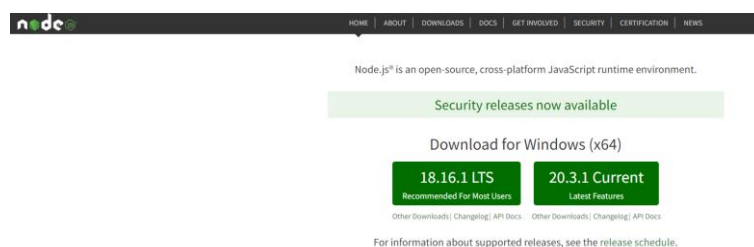
```
<input type="submit" id="applyBtn" value="Apply" style="background-color: aliceblue;
align-self: center;" width="2em">
```

4.1.2 Инсталирање Firebase и подешавање пројекта

Да би радили са Firebase-ом, морамо га инсталирати и на њему направити пројекат.

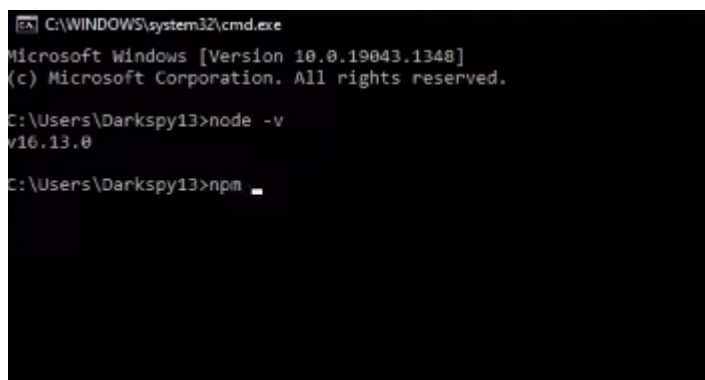
Први корак је инсталирање Node.js са следећег линка: <https://nodejs.org/en>

Одабрати најновију верзију и пратити кораке за инсталирање преко Setup Wizarda.



Слика 11 - Инсталирање Node.js

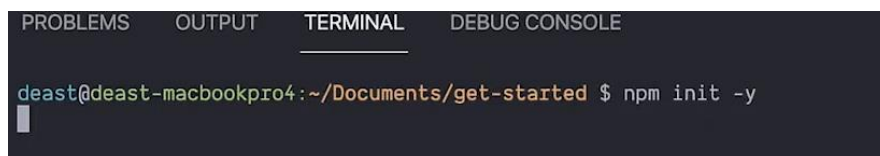
У терминалу се лако може проверити да ли је инсталација успешна уносом команде `node -v`



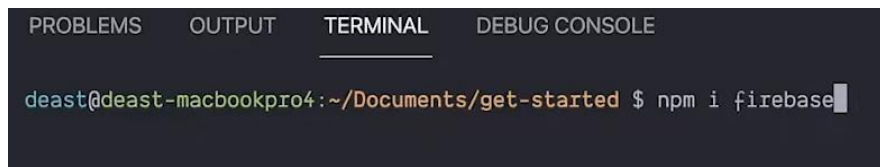
Слика 12 - Провера успешне инсталације

За инсталацију Firebase потребна су два корака:

```
npm init -y  
npm i firebase
```

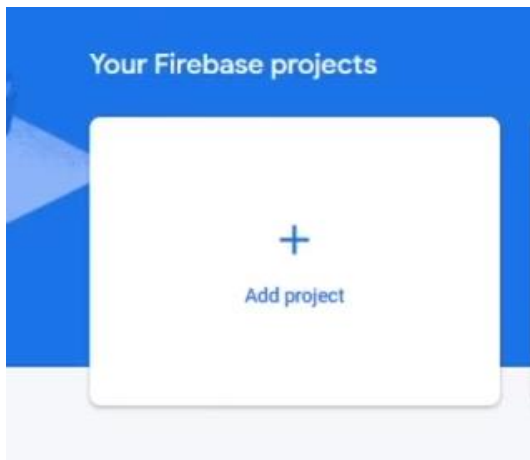


Слика 13 - Први корак у инсталацији Firebase



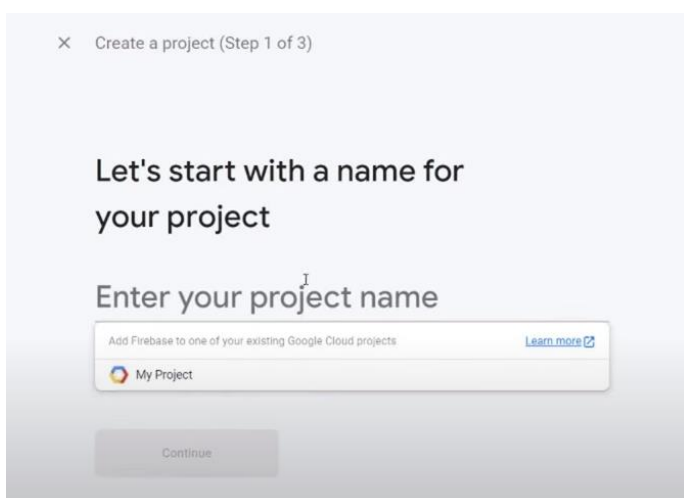
Слика 14 - Други корак у инстаалцији Firebase

За прављење новог пројекта пријавити се на свој Гугл налог и отићи на Firebase конзолу, затим кликнути Add project.

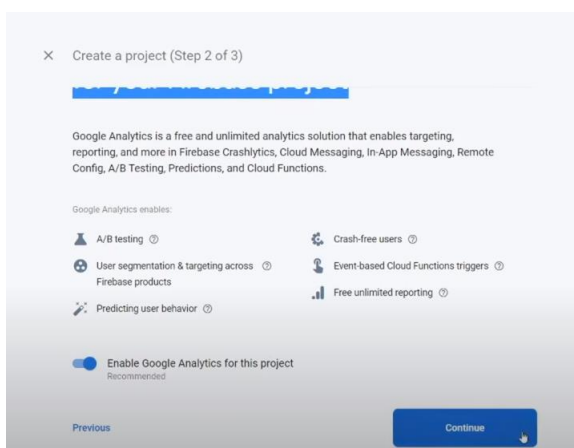


Слика 15 - Креирање новог пројекта

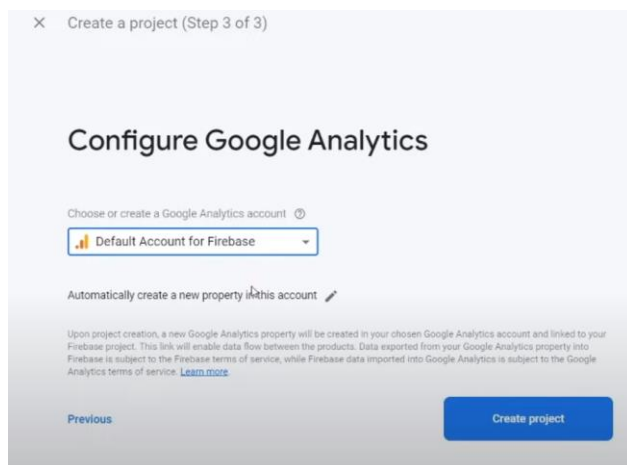
Дати свом пројекту име, дозволити Google Analytics (опционо) и одабрати подразумевани налог



Слика 16 - Први корак у креирању пројекта

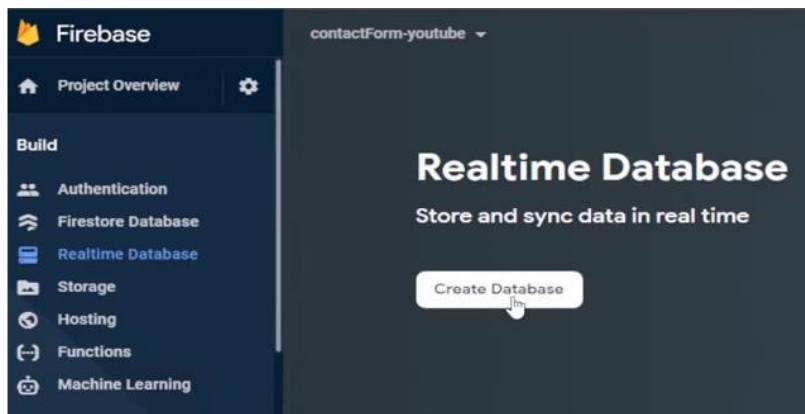


Слика 17 - Опционо дозвољавање Google Analytics



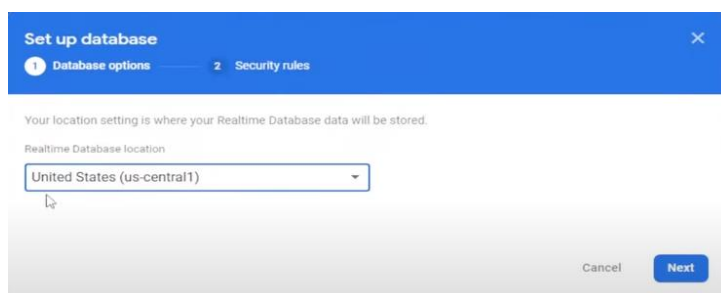
Слика 18 - Одабир подразумеваног (свог) налога

Са леве стране отићи на Realtime Database и направити нову базу

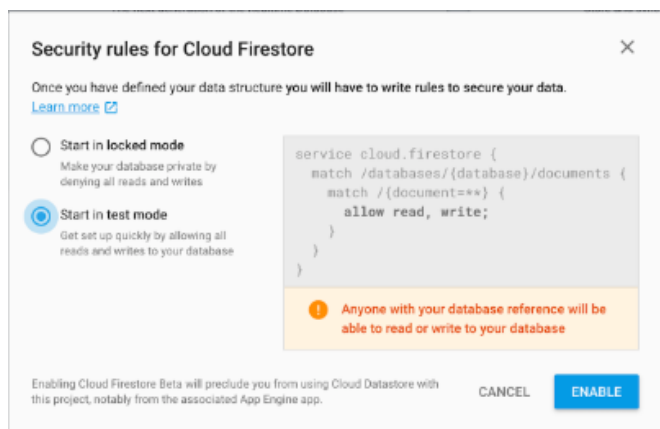


Слика 19 - Прављење нове базе у реалном времену

Одабрати на ком серверу ће база бити сачувана и сигурносна правила:



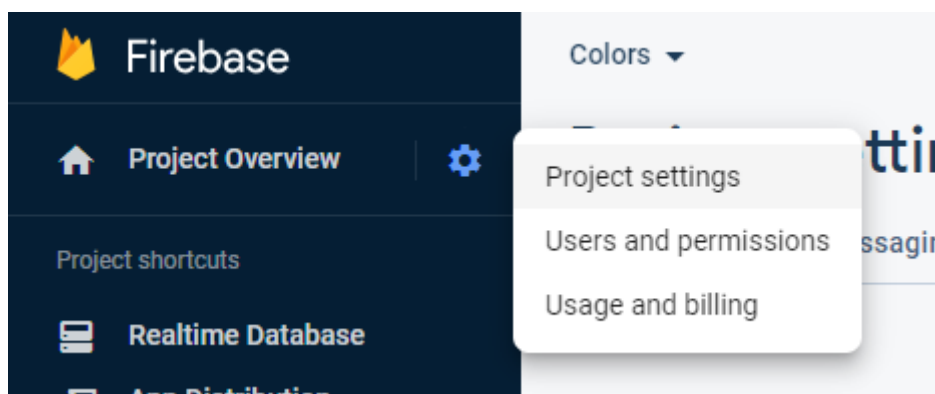
Слика 20 - Одабир сервера на ком ће база бити сачувана



Слика 21 - Одабир сигурносних правила

У пројекту сада треба додати своју веб апликацију.

У горњем левом углу притиснути иконицу зупчаника, па Project settings



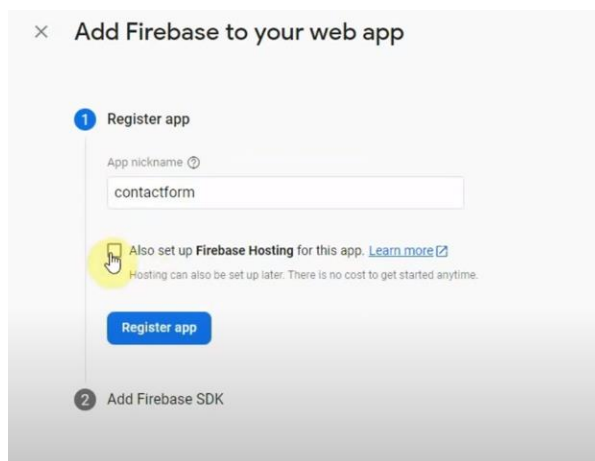
Слика 22 - Отварање подешавања пројекта у Firebase

Скроловати доле и на списку својих апликација притиснути Add app

Одабрати веб апликацију (иконица са </> симболом)

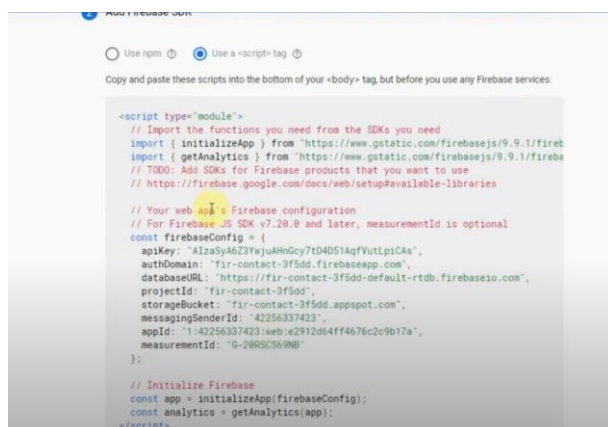


Слика 23 - Додавање нове веб апликације у пројекат



Слика 24 - Први корак у додавању базе

Када се први корак одради, појавиће се део код који треба копирати да би се база иницијализовала.



Слика 25 - Други корак у додавању базе

4.1.3 Функционалност веб странице у Јаваскрипту

Сада се може кренути са имплементацијом функционалности у Јаваскрипту. Код се куца у истом фајлу као и претходно направљена табела, али под <script> тагом.

Потребне библиотеке и иницијализација базе:

```
import { initializeApp from "https://www.gstatic.com/firebasejs/9.19.1/firebase-app.js";

import { getAnalytics } from "https://www.gstatic.com/firebasejs/9.19.1/firebase-analytics.js";

import {getDatabase, ref, set, child, update, remove, onValue} from
"https://www.gstatic.com/firebasejs/9.19.1/firebase-database.js";

const firebaseConfig = {
  apiKey: "AIzaSyCAZ1WbaIk0VXU2ihwfp9J8v4_zYRpyk38",
  authDomain: "colors-fe329.firebaseio.com",
  databaseURL: "https://colors-fe329-default-rtdb.europe-west1.firebaseio.com",
  projectId: "colors-fe329",
  storageBucket: "colors-fe329.appspot.com",
```

```

messagingSenderId: "63570994143",
appId: "1:63570994143:web:fb0a02c920c538fb585a6a",
measurementId: "G-7YWECHQ3C9"
};
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);
var db = getDatabase();

```

После тога правимо променљиве које одговарају радио дугмићима:

```
const rb5 = document.getElementById("r5");
```

Треба направити и променљиве које ће одогуварати бојама, али се оне не иницијализују преко id поља као променљиве радио дугмића. Њихове вредности ће бити додељене касније.

Даље су постављене референце које ослушкују промене у бази података да би приказ на сајту увек био исти као и у самој бази:

```

var bgd3Ref = ref(db, 'Patterns/Pattern3/backgroundColor');

onValue(bgd3Ref, (snapshot) =>{
    var data = snapshot.val();
    document.getElementById("3-1").value = data;
});

```

Кликом на дугме Apply промене треба да се упишу у базу.

Ослушкивач на клик дугмета:

```
$("#applyBtn").on("click", function({}));
```

Морамо и осигурати да је кориснички унос исправан шестоцифрени хексадецимални број. За то се користе регуларни изрази, помоћу којих се контролише да ли неки тескт одговара одређеним условима.

Израз који се користи за проверу исправности уноса:

```
var regex = /[0-9A-Fa-f]{6}/g;
```

Овај израз значи да су подржани карактери 0-9, a-f, A-F и да се појављују тачно 6 пута.

Сва поља у изабраном шаблону се пореде са изразом и ако је сваки унос исправан, можемо проследити те вредности променљивама за боје. Ако није, на екрану приказати одговарајућу поруку.

```

        if (rb4.checked && document.getElementById("4-1").value.match(regex) &&
document.getElementById("4-2").value.match(regex) && document.getElementById("4-
3").value.match(regex) && document.getElementById("4-4").value.match(regex) &&
document.getElementById("4-5").value.match(regex) && document.getElementById("4-
6").value.match(regex) && document.getElementById("4-7").value.match(regex))
{
    backgroundColor = document.getElementById("4-1").value;
    nonselectedColor = document.getElementById("4-2").value;
    textColor = document.getElementById("4-3").value;
    mainColor = document.getElementById("4-4").value;
    selectedColor = document.getElementById("4-5").value;
    progressColor = document.getElementById("4-6").value;
    otherColor = document.getElementById("4-7").value;
    num = 4;
}
else if (rb4.checked)
{
    alert("Inputs must be valid hexadecimal color of 6    characters");
}

```

Када смо сигурни да је унос исправан, можемо га уписати у базу:

```

update(ref(db, 'Patterns/Pattern' + num),{
    backgroundColor: backgroundColor,
    textColor: textColor,
    selectedColor: selectedColor,
    nonselectedColor: nonselectedColor,
    progressColor: progressColor,
    mainColor: mainColor,
    otherColor: otherColor,
    isSelected: true
});

```

На крају додајемо осигуравање да је само један шаблон одабран, што ће бити важно у Андроид Студију:

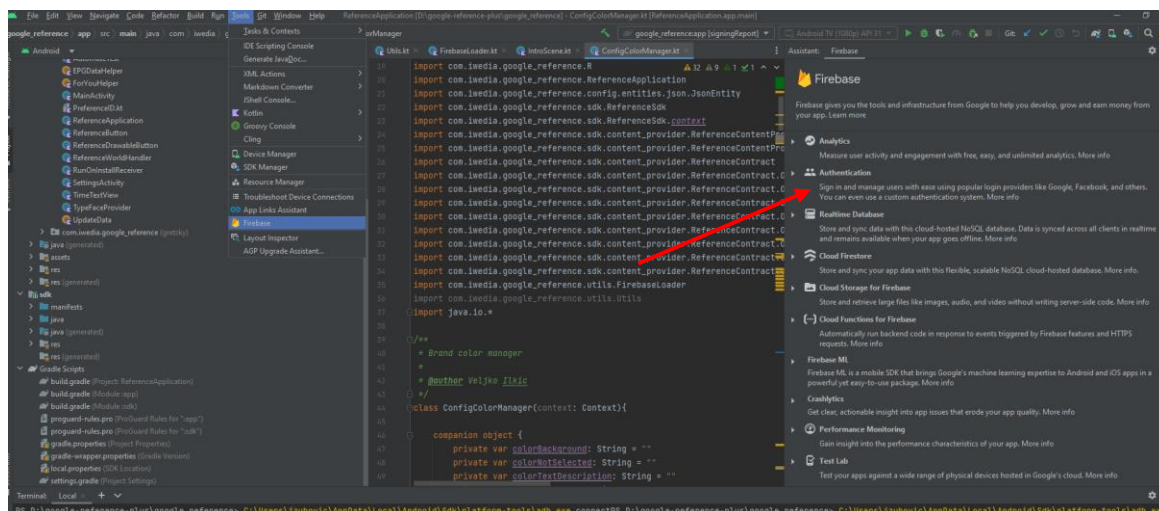
```

for (var i = 1; i<6; i++){
    if (i!=num) update(ref(db, 'Patterns/Pattern' + i),{
        isSelected:false
    })
}

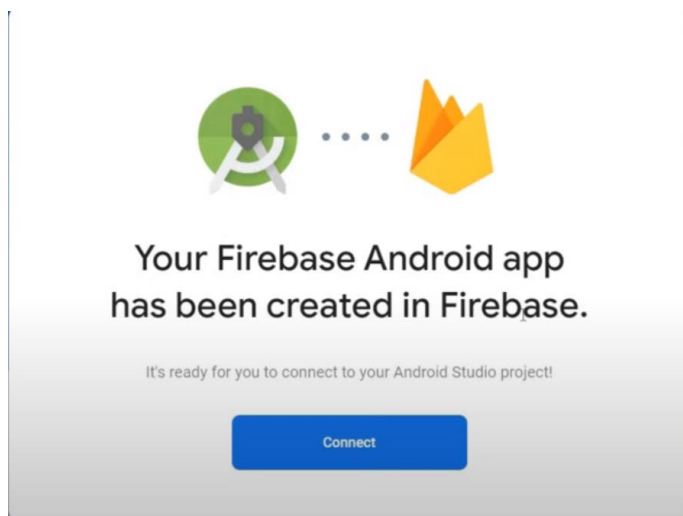
```

4.2 Програмско решење у Андроид Студију

У Андроид Студију се прате кораки као и у HTML, са разликом да се база додаје једноставно кликом на Tools->Firebase->Realtime Database. Одатле Андроид Студио води корисника на страницу за пријављивање на свој налог, где корисник просто треба да се пријави и одабере свој пројекат. Сви остали кораци се раде аутоматски.



Слика 26 - Додавање Firebase у Андроид Студио



Слика 27 - Последњи корак у повезивању Firebase са Андроид Студијом

У коду сада треба иницијализовати базу:

```
Firestore.initialize(context)
```

Као и у HTML, постављањају референце на поља базе:

```
var prg2Ref : DatabaseReference =  
FirestoreDatabase.getInstance().getReference("Patterns/Pattern2/progressColor")
```

За разлику од веб странице, не треба ослушкивати све промене одједном, већ треба проверити који шаблон је одабран и учитати боје боје само тог шаблона. Зато се прво проверава `isSelected` поље за које је на самом крају поглавља 4.1. наведено да ће бити важно у Андроид Студију. Када се то провери, проверавају се остала поља.

Провера одабраног шаблона, ослушкивање и примена промене

```
isslc1Ref.addValueEventListener(object : ValueEventListener {
    override fun onDataChange(dataSnapshot: DataSnapshot) {
        if (dataSnapshot.getValue().toString() == "true"){
            bgd1Ref.addValueEventListener(object : ValueEventListener
            {
                override fun onDataChange(dataSnapshot: DataSnapshot) {
                    colorBackground = '#' +
                    dataSnapshot.getValue().toString()
                }
            }
            override fun onCancelled(error: DatabaseError) {
                // Failed to read value
                Log.w(ContentValues.TAG, "Failed to read value.",
                error.toException())
            }
        })
    }
})
... }
```

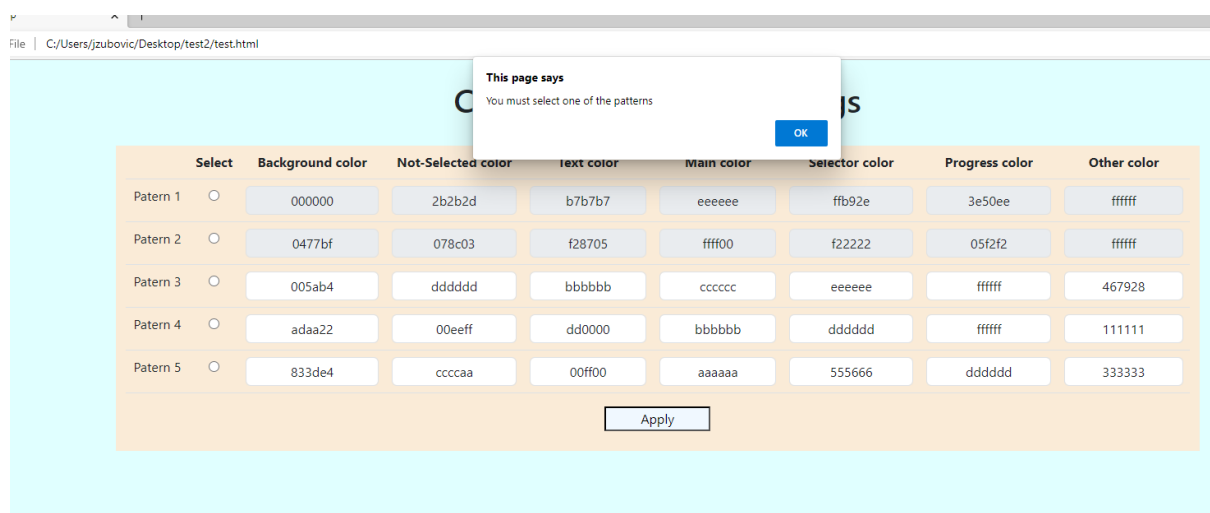
5. Тестирање и верификација

Тестирање и верификација су урађени на случајевима приказаним у табели 1. За тестирање апликације коришћена је претходно направљена веб страница, за проверу да ли се боје исправно мењају Google TV, а за мерење времена протеклог од потврде уноса у апликацији до стварне промене Андроид Студио.

Тестни случај:	Коришћено окружење:
Тест без одабраног шаблона	Веб страница
Унос мање од 6 карактера	Веб страница
Унос карактера који није валидна хексадецимална цифра	Веб страница
Провера да ли се боје мењају како треба	Google TV
Тест протеклог времена од клика на дугме за потврду до примене промена	Андроид Студио

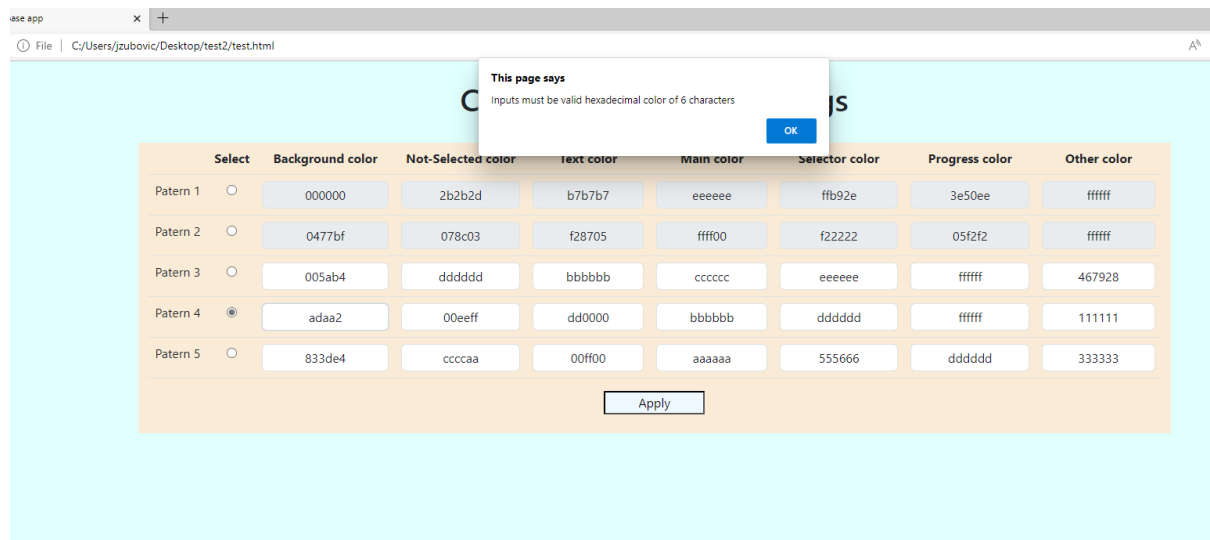
Табела 1 - Тестни случајеви

За почетак, веб апликацији притицнути дугме за упис у базу без одабира неког од шаблона. Страница ће избацити обавештење да један од шаблона мора бити одабран.

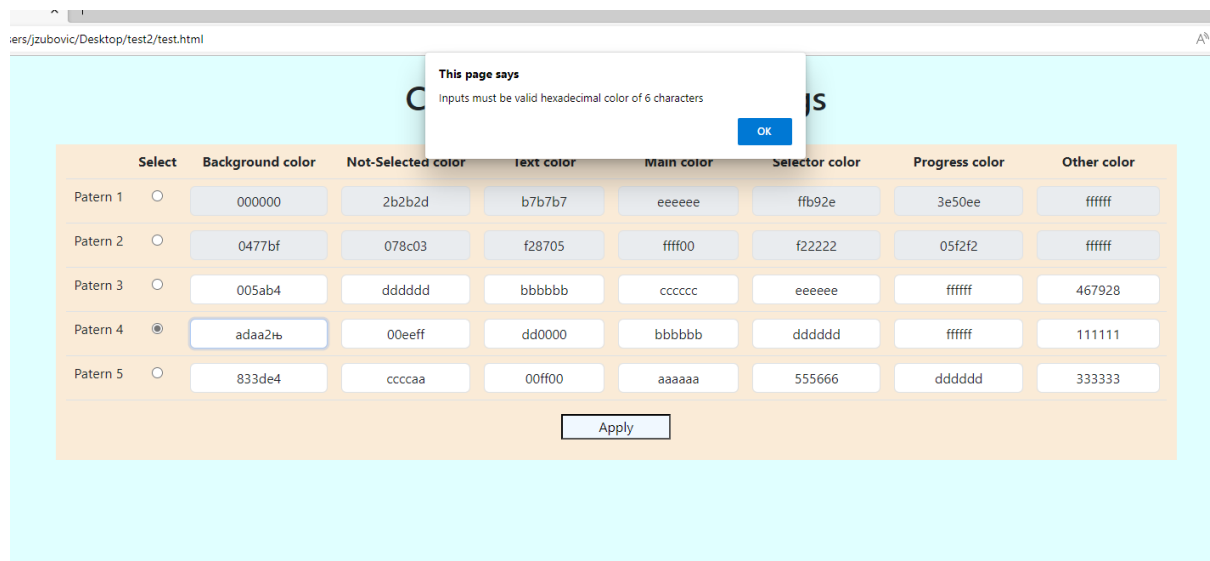


Слика 28 - Тест без одабраног шаблона

Онда уписати у једно од поља одабраног шаблона мање од 6 карактера или неки карактер који није хексадецимална цифра (валидне су цифре 0-9 и мала и велика латинична слова а, б, ц, д, е и ф). Страница ће избацити обавештење да унос мора бити валидан хексадимални низ дужине 6 карактера. Уношење више од 6 карактера није омогућено.



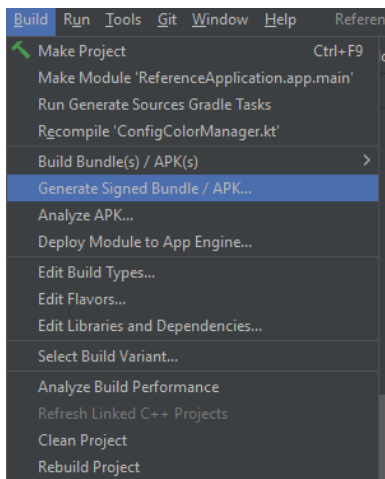
Слика 29 - Унос мање од 6 карактера



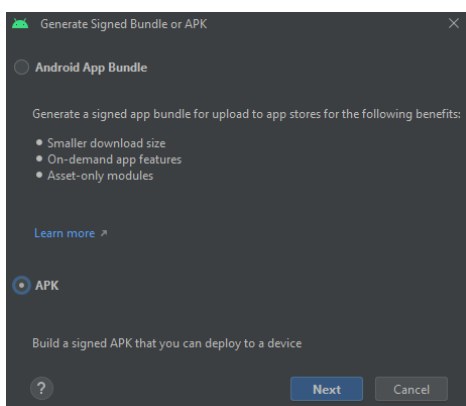
Слика 30 - Унос карактера који није валидна хексадецимална цифра

Проверити да ли су се унете вредности исправно уписале у Firebase.

Што се тиче Андроид Студија, да би се проверило да ли се боје исправно мењају потребно је превести код, направити излазни фајл са .apk екстензијом и инсталирати тренутну верзију апликације на сет-топ-бокс. Превођење и генерисање се врши уз помоћ кључа и то на следећи начин:

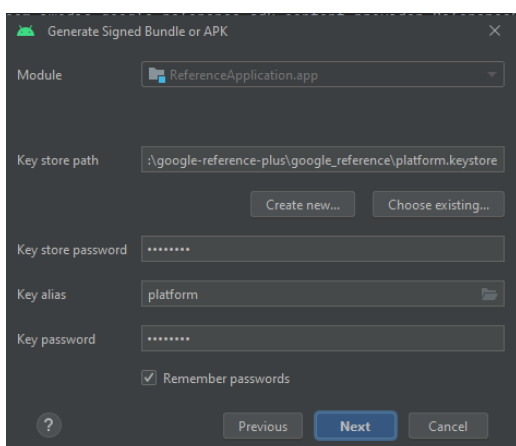


Слика 31 - Генерисање излазног фајла помоћу кључа



Слика 32 - Одабир екстензије излазног фајла

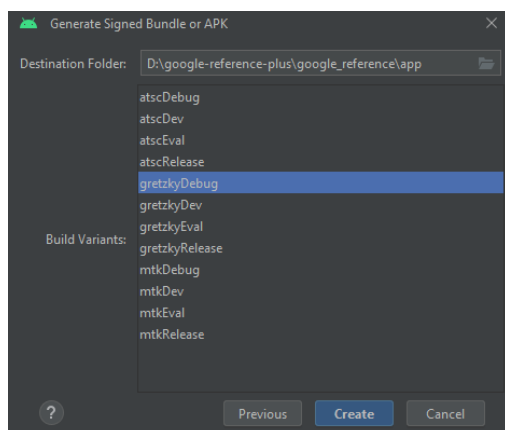
Потребно је изабрати .apk екстензију јер се .aab користи за коначну верзију која се поставља на Google Play, а не за тестирање



Слика 33 - Попуњавање података о кључу

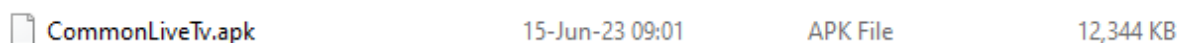
Што се тиче кључа, препоручује се коришћење већ постојећег, а остале податке попунити уз помоћ некога ко је са тим кључем већ радио

На крају треба одабрати где ће излазни фајл бити смештен. За тестирање, то ће бити Debug фолдер, а за крајњу верзију Release фолдер.



Слика 34 - Одабир фолдера где ће бити смештен излазни фајл

На одабраној путањи ће се појавити фајл са аутоматски генерисаним именом. Потребно га је преименовати у CommonLiveTv.apk



Слика 35 - Преименовани излазни фајл

У терминалу Андроид Студија се позиционирати путању излазног фајла и унети следеће команде:

```
C:\Users\<username>\AppData\Local\Android\Sdk\platform-tools\adb.exe
connect <ipadress>
C:\Users\<username>\AppData\Local\Android\Sdk\platform-tools\adb.exe
install <pathToDestinationFolder>\CommonLiveTv.apk
```

```
PS D:\google-reference-plus\google_reference> C:\Users\jzubovic\AppData\Local\Android\Sdk\platform-tools\adb.exe connect 192.168.234.58
connected to 192.168.234.58:5555
PS D:\google-reference-plus\google_reference> C:\Users\jzubovic\AppData\Local\Android\Sdk\platform-tools\adb.exe install D:\google-reference-plus\google_reference\app\gretzky\debug\CommonLiveTv.apk
Performing Streamed Install
Success
```

Слика 36 - Инсталирање на сет-топ-бокс ради тестирања читања базе у Андроид Студију

Укључити сет-топ-бокс и на њему своју апликацију.

Ако се након инсталације боје мењају као што је унето у апликацији, све је у реду.

Важно је напоменути да се промена неће приказати ако је мени на телевизору већ укључен. Прво треба изаћи из менија и поново ући.

У току дебаговања се може користити логовање у логкет Андроид Студија.

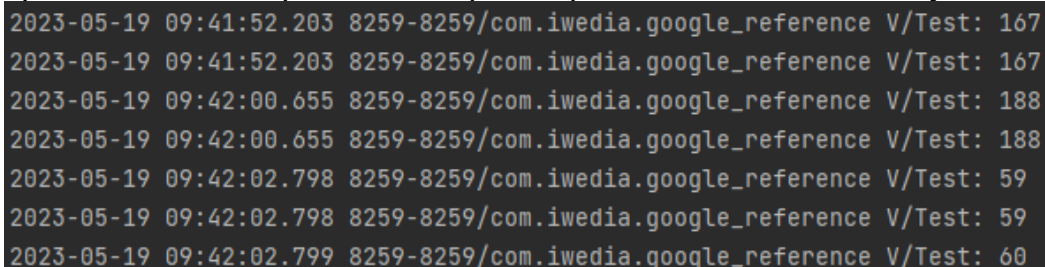
У следећем тесту је проверено колико времена прође од клика на дугме за потврду промена на веб страници до имплементације на ТВ-у. Зато је додата следећа линија кода у HTML:

```
update(ref(db, 'Patterns/Pattern' + num),{time:Date.now()});
```

У Андроид Студио је затим додат следећи код:

```
time5Ref.addValueEventListener(object: ValueEventListener {  
    override fun onDataChange(dataSnapshot: DataSnapshot) {  
        val timeFB = dataSnapshot.getValue().toString().toLong()  
        val timeAS = System.currentTimeMillis()  
        Log.v("Test", (timeAS - timeFB).toString())  
    }  
})
```

Време промене је зависно од тренутног квалитета мреже, али ако нема озбиљних проблема са везом промена се изврши за ред величине 100 милисекунди.



```
2023-05-19 09:41:52.203 8259-8259/com.iwedia.google_reference V/Test: 167  
2023-05-19 09:41:52.203 8259-8259/com.iwedia.google_reference V/Test: 167  
2023-05-19 09:42:00.655 8259-8259/com.iwedia.google_reference V/Test: 188  
2023-05-19 09:42:00.655 8259-8259/com.iwedia.google_reference V/Test: 188  
2023-05-19 09:42:02.798 8259-8259/com.iwedia.google_reference V/Test: 59  
2023-05-19 09:42:02.798 8259-8259/com.iwedia.google_reference V/Test: 59  
2023-05-19 09:42:02.799 8259-8259/com.iwedia.google_reference V/Test: 60
```

Слика 37 - Приказ протеклог времена до примене промена у милисекундама

6. Могућа унапређења

Тренутно решење је намењено мрежним администраторима. Постоји могућност додавања опције крајњим корисницима да сами мењају боје. Тада би било пожељно променити уносе у табели да не буду текстуална поља која очекују исправан хексадецимални унос већ палете са бојама. Разлог за то је чињеница да врло мало људи разуме како хексадецимални запис боје ради и треба им омогућити нешто што разумеју.

База се такође може проширити тако да не чува само шаблоне са бојама, већ и текстуалне исписе, слике, логое и тако даље. Ово је корисно јер се са Андроид Студијом може повезати само једна Firebase база по пројекту, и ако постоји потреба за мењањем још неке ставке у реалном времену то не би било могуће са другом Firebase базом.

Сва функционалност се може пребацити и у посебну класу. Када се мења само једна ставка, као што су боје у овом раду, то није неопходно, али када се мења неколико ставки много је прегледније и лакше за одржавање имати све функције везане за Firebase на једном месту.

7. Закључак

Приказано је једно решење мењања боја Google TV-а преко Firebaseове базе података у реалном времену.

Направљена је веб страница која садржи табелу за унос боја, и у њој омогућено уписивање вредности поља табеле у базу уз одговарајућу контролу да је унос исправан.

Након тога је у изворном коду апликације додата функција која вредности из табеле чита у реалном времену и примењује их. На стварном Google TV-у је дата функционалност и тестирана.

8. Литература

1. https://en.wikipedia.org/wiki/Television_set - О телевизији уопште, мај 2023.
2. https://en.wikipedia.org/wiki/Smart_TV - О паметној телевизији, мај 2023.
3. <https://en.wikipedia.org/wiki/HTML> - О HTML, мај 2023.
4. <https://en.wikipedia.org/wiki/CSS> - О CSS, мај 2023.
5. Џон Дакет - HTML and CSS: Design and Build Websites, John Wiley & Sons Inc, 2011, ISBN 1118008189 - Књига о HTML и CSS
6. [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)) - О Андроиду, мај 2023.
7. <https://www.rt-rk.uns.ac.rs/?q=predmeti/e2/pnrs-1-projektovanje-namenskih-ra%C4%8Dunarskih-struktura-1> - Сајт предмета Пројектовање наменских рачунарских система 1 (Пројектовање и архитектура рачунарских система - Системи засновани на андроиду по новој акредитацији), мај 2023.
8. <https://firebase.google.com/docs> - Firebase документација, мај 2023.
9. <https://getbootstrap.com> - Бутстреп документација, април 2023.
10. <https://htmldog.com/guides/javascript/advanced/regex/> - О регуларним изразима, јун 2023.
11. <https://developer.mozilla.org/en-US/docs/Web/HTML> - HTML документација, април 2023.
12. <https://developer.android.com/studio> - Андроид Студио документација, мај 2023.