

NADOGRADNJA MREŽNE APLIKACIJE ZA PRETRAŽIVANJE PODATAKA O OPAŽANJIMA TELESKOPIMA MAGIC

Kruhoberec, Matej

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj
Strossmayer University of Osijek, Department of Physics / Sveučilište Josipa Jurja
Strossmayera u Osijeku, Odjel za fiziku**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:160:839523>

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

Download date / Datum preuzimanja: **2025-02-18**



Repository / Repozitorij:

[Repository of Department of Physics in Osijek](#)



SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ODJEL ZA FIZIKU

Matej Kruhoberec

**NADOGRADNJA MREŽNE APLIKACIJE ZA PRETRAŽIVANJE
PODATAKA O OPAŽANJIMA TELESKOPIMA MAGIC**

DIPLOMSKI RAD

Osijek, listopad 2020.

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

ODJEL ZA FIZIKU

Diplomski studij fizike i informatike - nastavnički smjer

**NADOGRADNJA MREŽNE APLIKACIJE ZA PRETRAŽIVANJE
PODATAKA O OPAŽANJIMA TELESKOPIMA MAGIC**

DIPLOMSKI RAD

Predmet: Diplomski rad

Mentor: doc. dr. sc. Dario Hrupec

Student: Matej Kruhoberec

Matični broj: 612

Osijek, listopad 2020.

SAŽETAK

Cilj ovog diplomskog rada je unaprijediti već postojeću mrežnu aplikaciju MAGIC Legacy (<http://lapp.irb.hr/MAGIClegacy>). Aplikacija se nalazi na serveru Instituta Ruđer Bošković, a dostupna je samo članovima kolaboracije MAGIC. Upotrebljava se za pretraživanje podataka o opažanjima teleskopima MAGIC. Teleskopi MAGIC se koriste za opažanje visokoenergijskih gama-izvora koji su i u današnje vrijeme i dalje prepuni neodgovorenih pitanja. Aplikacija je funkcionalna, no javila se potreba za značajnim dopunama. Način na koji se unose početni podaci, koordinate oko kojih pretražujemo nebo, bio je ograničen za korisnika. Cilj je unaprijediti aplikaciju na način da se podaci mogu upisati bez potrebe korištenja drugih alata te dodati najnoviji katalog nebeskih objekata koje je opažao satelit *Fermi*. Za rješenje tog problema u ovom su radu korišteni razni alati. Najvažniji je naravno programski jezik Python i njegov paket programskih alata Astropy koji je uvelike olakšao posao. Nadogradnja je odrađena u planiranom obliku te je implementirana i stavljena na korištenje astronomskoj zajednici kolaboracije MAGIC.

SUMMARY

The main goal of this thesis is to upgrade the existing network application MAGIC Legacy (<http://lapp.irb.hr/MAGIClegacy>). The application is hosted by the Ruđer Bošković Institute servers and available only to the members of the MAGIC collaboration. The application is used for browsing data about MAGIC telescopes observations. The MAGIC telescopes are used for observing high energy gamma emitters which are to this day full of unanswered questions. The application is functional but there is a need for significant upgrades. The initial data input, sky coordinates, was limited for the user. The goal is to upgrade the application in a way that the data input can be done without the need for additional tools and to add the newest catalog of celestial objects observed by the *Fermi* satellite. This

thesis uses various tools to solve the problem. The most important is, of course, Python programming language and its Astropy utility bundle which significantly made the job easier. The upgrade is done in the planned form and it is implemented and put to use for the members of the astronomy society of the MAGIC collaboration.

Sadržaj

1. Uvod.....	7
2. Gama-astronomija.....	8
3. Teleskopi MAGIC	9
4. Nebeski koordinatni sustavi.....	11
4.1 Galaktični koordinatni sustav	11
4.2 ICRS – International Celestial Reference System	12
5. Astropy	13
5.1 Klase SkyCoord i Angle	13
6. Mrežna aplikacija: MAGIC Legacy	14
7. Datoteke potrebne za rad aplikacije	15
8. Dio aplikacije napisan u programskom jeziku Python	16
9. Implementacija četvrtog <i>Fermi</i> -kataloga	17
10. Implementacija mogućnosti unosa koordinata na različite načine.....	21
11. Modifikacija index.html datoteke i dodavanje javascript skripte++	24
12. Završetak nadogradnje	25
13. Buduće nadogradnje	26
14. Zaključak	27
16. Prilozi.....	30
16.1 index.html datoteka.....	30
16.2 find.py datoteka	32
16.3 main.js datoteka.....	38

1. Uvod

MAGIC Legacy je mrežna aplikacija koja se nalazi na serveru Instituta Ruđer Bošković. Koristi se za pretragu podataka o opažanjima teleskopima MAGIC. Aplikacija je jedan od rezultata diplomskog rada Jelene-Kristine Željeznak. [12] Aplikacija za ulazne parametre uzima koordinate u ICRS okviru, polumjer udaljenosti kako bi definirali dio neba koji pretražujemo te izbor baza podataka iz kojih pretražujemo. Koordinate se prije nadogradnje može upisati samo u ICRS okviru (rektascenzija u satnom obliku, deklinacija u stupnjevima). Cilj rada je omogućiti unos koordinata i u galaktičkom obliku, omogućiti unos koordinata u ostalim tipovima kuteva te dodati četvrti katalog objekata satelita *Fermi* kao izvor za pretraživanje objekata. U rješenju zadatka koristiti će se programski jezici: Python, Javascript, HTML, CSS. Osim programskih jezika potrebni su i alati fv, Notepad++, internet preglednik te jedan od poslužitelja lokalnog servera kako bi se aplikacija mogla testirati tijekom razvojnog procesa.

2. Gama-astronomija

Gama-zračenje su elektromagnetski valovi vrlo visoke energije. Mogu nastati prilikom kvantnih prijelaza atomskih jezgri s pobuđenih stanja, u procesu zakočnog zračenja brzih elektrona prilikom sudara s atomskim jezgrama, prilikom inverznog Comptonovog raspršenja, pri nastanku piona, pri nastanku para foton-foton, sinkrotronskim zračenjem i u procesima pretvorbi subatomske čestice. Maria Curie-Sklodowska i Pierre Curie su ih otkrili 1898. pri proučavanju prirodne radioaktivnosti. [1]

Načini na koji se vidi interakcija gama-zraka s materijom su: fotoelektrični efekt, Comptonovo raspršenje i tvorba parova. [8]

U zadnjih 20-ak godina gama-astronomija je postala moćan alat u proučavanju fizike kozmičkih gama-zraka. Galaktička i izvangalaktička magnetska polja ne utječu na smjer fotona, njihovi smjerovi dolaska i energije donose informacije o izvorima te su jednostavniji za detekciju od neutrina. Detekcija visokoenergijskih gama-zraka je zadnji korak u korištenju cijelog elektromagnetnog spektra za istraživanje procesa i fenomena u svemiru. Gama-zrake dolaze iz raznih kozmičkih izvora: supernova, neutronske zvijezde, kvazara, aktivnih galaktičkih jezgara. Proučavanje visokih i vrlo visokih energija gama-zraka nam pomaže razumjeti procese ubrzanja nabijenih čestica do najviših energija.

Gama-astronomija bi također mogla dati tragove novih otkrića u kozmologiji opažanjima infracrvenog pozadinskog zračenja i traženjem primordijalnih crnih rupa. Mogla bi proširiti razumijevanje fizike čestica izvan granica trenutnog standardnog modela. [2]

Godine 1961. satelit „Explorer XI“ je prvi detektirao gama-zrake iz svemira. Prvi značajni rezultati su dobiveni 1972. i 1973. kada je satelit SAS-2 sedam mjeseci skupljao podatke od oko 8000 fotona energije od 30 MeV do 5 GeV. U periodu između 1975. i 1982. satelit COS-B je identificirao 25 galaktičkih izvora fotona

na GeV razinama energije. Pred kraj 60-ih talijanski fizičar Cocconi je predložio ideju da se fotoni energijske razine TeV istražuju pomoću detektora na Zemljinoj površini. Indirektna mjerenja VHE („Very high energy“) i UHE („Ultra high energy“) fotona su ostvarena 70-ih prvim Čerenkovim teleskopima. Na Krimu 1972. Čerenkovljev teleskop je otkrio roj fotona u području energija TeV, koji je dolazio iz smjera binarnog izvora Cygnus X-3. Time je službeno započela era VHE gama-astronomije. [2]

Drastičan napredak donio je EGRET (Energetic Gamma Ray Experiment Telescope) koji se nalazio na satelitu CGRO (Compton Gamma Ray Observatory) te se koristio od 1991. do 2000. godine. Satelit INTEGRAL (INTERNational Gamma-ray Astrophysics Laboratory) lansiran je 2002. godine, a satelit SWIFT lasiran je 2004. godine. Oba satelita su još uvijek u upotrebi. Satelit *Fermi* (Gamma-ray Large Area Space Telescope) lansiran je 2008. godine te je predstavnik nove generacije gama-satelita. Do tada su sateliti imali komore na iskre, dok je *Fermi* opremljen segmentiranim silicijskim detektorom te je podigao gornju energijsku granicu sa približno 20 GeV na 300 GeV.

3. Teleskopi MAGIC

MAGIC je sustav dvaju Čerenkovljevih teleskopa. MAGIC-I je operativan od 2004. Konstrukcija MAGIC-II je završila 2009., od onda su oba teleskopa u upotrebi s kratkom pauzom 2012. zbog nadogradnje. [3]



Slika 1 – Teleskopi MAGIC na Kanarskim otocima (Španjolska) [3]

Slika prikazuje dva MAGIC teleskopa ukupne površine svakog od reflektora 236 m². Opservatorij se nalazi na Kanarskim otocima na otprilike 2200 m nadmorske visine. Zbog geografskog položaja i nadmorske visine Kanarski otoci su odlično mjesto za astronomska opažanja. [3]

Svemir i njegov razvoj proučavamo opažanjem elektromagnetskih valova, kozmičkog zračenja, neutrina i u novije vrijeme gravitacijskih valova. Spektar elektromagnetnog zračenja se prostire od radiovalova energije 10⁻⁹ eV do vrlo visokih energija gama-zračenja energija 100 TeV. IACT (Imaging Atmospheric Cherenkov Telescope) su teleskopi koji se nalaze na Zemljinoj površini i služe za opažanje VHE elektromagnetskih valova. Iako se visokoenergetski gama-kvanti apsorbiraju u atmosferi može ih se opaziti indirektno. Interakcijom primarne gama-zrake u atmosferi nastaje kaskada sekundarnih visokoenergijskih čestica. [3] Čerenkovljeva metoda se temelji na Čerenkovljevom zračenju. Čerenkovljevo zračenje nastaje kada se čestica giba kroz dielektrično sredstvo i brzina joj je veća nego brzina svjetlosti u tom sredstvu. Zračenje se emitira pod točno određenim kutem koji ovisi o indeksu loma sredstva. Kada nabijena čestica prođe kroz dielektrično sredstvo dogodi se interakcija s molekulama u blizini te čestice. Naruši se električna neutralnost molekula i dolazi do polarizacije koja se naizmjenično pojavljuje i nestaje usporedno s prolaskom nabijene čestice kraj molekula. U toj se promjeni polarizacije molekula otpušta zračenje. [8] Čerenkovljevi fotoni imaju energije u vidljivom i UV (ultraljubičastom) dijelu spektra, prolaze kroz atmosferu te ih se može detektirati na površini Zemlje dovoljno osjetljivim instrumentima. [3]

IACT su detektori za visokoenergijske gama-zrake koji se nalaze na Zemljinoj površini. Posjeduju zrcalo koje skuplja svjetlost i fokusira ga na kameru. Zbog toga barem na prvi pogled podsjećaju na optičke teleskope. Detektiraju svjetlost koja nastaje Čerenkovljevom efektom u dijelu spektra između plave boje i UV zračenja. Čerenkovljev efekt se javlja kad se relativističke čestice gibaju brže od

svjetlosti u atmosferi (zraku). IACT detektiraju Čerenkove fotone za jedan slučaj gama-zrake na vrhu atmosfere. Detektiraju se kamerom kao karakteristična slika koja omogućava identifikaciju zračenja kao zračenje koje je uzrokovala gama-zraka te određuju energiju i smjer iz kojeg zraka dolazi. [4]

4. Nebeski koordinatni sustavi

Koordinatni sustavi nam omogućuju opisivanje položaja točke na krivulji, pravcu, plohi, u ravnini ili prostoru pomoću brojeva – koordinata. [5]

Postoje različiti koordinatni sustavi: Kartezijev, sferni, ekvatorski itd. Od astronomskih koordinatnih sustava najčešće se upotrebljavaju galaktični koordinatni sustav i ICRS. ICRS je akronim za „International Celestial Reference System“ (Međunarodni nebeski referentni sustav). U svim koordinatnim sustavima postoje osi ili kutevi koji jednoznačno određuju položaj točke u prostoru. Iako su galaktični i ICRS sustavi namijenjeni za određivanje položaja nebeskih tijela u prostoru, među njima naravno postoje razlike.

4.1 Galaktični koordinatni sustav

Kao što mu samo ime govori, za pomoć pri određivanju položaja nebeskih tijela koristimo našu galaksiju Mliječni put. Galaktičke koordinate su galaktička dužina i galaktička širina. Pomoću te dvije koordinate možemo odrediti položaj bilo kojeg tijela na nebeskoj sferi. Galaktička širina se mjeri u stupnjevima sjeverno ili južno od galaktičke ravnine. Kut je pozitivan za tijela koja se nalaze prema sjevernom galaktičkom polu i nalazi se u rasponu od 0 do 90 kutnih stupnjeva. Negativan je za tijela koja se nalaze između galaktičke ravnine i južnog galaktičkog pola i nalazi se u rasponu od 0 do -90 kutnih stupnjeva. Galaktička ravnina je određena galaktičkim ekvatorom. Galaktička ravnina je zamišljena ravnina koja prolazi simetrijom galaktike koja ju dijeli na dva dijela;

gornji i donji. Galaktički ekvator se nalazi pod 63 kutnih stupnjeva i 36 kutnih minuta u odnosu na nebeski ekvator. Nebeski ekvator je projekcija zemljinog ekvatora na nebesku sferu. Galaktička dužina se mjeri u stupnjeva istočno od zamišljene linije koja spaja Zemlju sa središtem galaktike (nalazi se u zviježđu Strijelac). Nalazi se u rasponu od 0 do 360 kutnih stupnjeva uzduž galaktičke ravnine. Prije 1958. galaktička dužina je bila proizvoljna točka postavljena u sjecištu galaktičkog i nebeskog ekvatora. Razvojem radioastronomije moglo se točnije odrediti položaj središta galaksije, te se od tada više ne koristi ta proizvoljna točka. Sjeverni galaktički pol leži u zviježđu Berenikina kosa a južni u zviježđu Kipar. Galaktička širina se označava slovom $-b$, galaktička dužina se označava slovom $-l$. [6]

4.2 ICRS – International Celestial Reference System

Međunarodna zajednica astronoma je 1997. odlučila da će se od 1. 1. 1998. ICRS koristiti kao referentni sustav za određivanje položaja nebeskih tijela. Ta odluka je donešena zato što ICRS koristi stotine radio-izvora izvan naše galaksije, najviše kvazara za postavljanje referentnog okvira. Ti kvazari su nam toliko daleko da nam se čini kao da su stacionarni, te kretanje Zemlje, Sunčevog sustava i Mliječnog puta ne izaziva paralaksu. ICRS kao i galaktički koordinatni sustav koristi dvije koordinate, međutim nisu istog naziva kao u galaktičkom koordinatnom sustavu. ICRS koristi rektascenziju i deklinaciju. Rektascenzija se izražava u satima, minutama i sekundama, a nalazi se na nebeskom ekvatoru. Poprima vrijednost 0h u točki gdje se Sunce nalazi prvog dana proljeća pa do 24h povećavajući se u smjeru istoka. Deklinacija poprima vrijednosti od 0 do +90 kutnih stupnjeva za tijela koja se nalaze sjeverno od nebeskog ekvatora, a od 0 do -90 kutnih stupnjeva za tijela koja se nalaze južno od nebeskog ekvatora. S obzirom da se položaji nebeskih tijela stalno prividno kreću zbog gibanja Zemlje, Sunčevog sustava i galaksije općenito, važno je uzeti fiksne točke za smjerove sjevernog i južnog pola. [7]

5. Astropy

Astropy je zajednički projekt astronomske zajednice. Cilj je razvijati jezgroviti paket za astronomiju koristeći programski jezik Python, unaprijediti funkcionalnost i olakšati upotrebu paketa sa drugim astronomskim paketima. Paket je prvenstveno namijenjen profesionalnim astronomima i astrofizičarima, no može biti koristan bilo kome tko želi razviti neki astronomski program. Uz Astropy Project dolaze i drugi paketi. Te druge pakete ne razvija nužno isti tim kao i glavni paket, međutim cilj razvijanja tih paketa je isti kao i za Astropy. Osim samog koda, Astropy je i zajednica astronoma i ljudi koji se bave astronomijom. Svi se slažu da je dijeljenje alata dobro za zajednicu i znanost. Zajednica je ta koja je zapravo osnova Astropyja.

Jezgra Astropyja (Astropy core package) sadrži razne klase, funkcije i okvir za pakiranje kako bi omogućio korištenje najčešćih astronomskih alata. Jezgra se sastoji od mnogobrojnih paketa od kojih svaki imaju svoju svrhu. [9]

5.1 Klase SkyCoord i Angle

Skycoord je klasa koja dolazi u sklopu Astropy paketa. Nju sam izdvojio jer će nam uvelike olakšati budući posao. SkyCoord klasa prima širok spektar ulaznih parametara za stvaranje SkyCoord objekta. Minimum podataka koji se mora dodati su jedna ili više nebeskih koordinata s jednoznačnim mjernim jedinicama. Ulaz mogu biti skalari, liste, nizovi. Također se može odrediti u kojem koordinatnom sustavu želimo upisati koordinate. Ako se ništa ne odredi, osnovni koordinatni sustav u kojem se stvara objekt je ICRS. [11]

Osim SkyCoord objekta koristi se i klasa naziva Angle. To je klasa koja instancira objekt koji predstavlja kut. Objektu se dodjeljuje iznos i tip kuta. Angle objekti odlično funkcioniraju u kombinaciji sa SkyCoord objektima s obzirom da dolaze iz istog programskog paketa.

6. Mrežna aplikacija: MAGIC Legacy

Mrežna aplikacija MAGIC Legacy nalazi se na serveru Instituta Ruđer Bošković. Upotrebljava se za pretragu podataka koji stižu sa teleskopa MAGIC.



MAGIC Legacy

[1] Search in a radius (r) around a position on the celestial sphere (RA, DEC)

Right ascension (HH:MM:SS) : : (range: $0 \leq \text{HH} < 24$, $0 \leq \text{MM} < 60$, $0.00 \leq \text{SS} < 60.00$)

Declination (DD:MM:SS) : : (range: $-90 \leq \text{DD} \leq +90$, $0 \leq \text{MM} < 60$, $0.00 \leq \text{SS} < 60.00$)

Radius of interest (r) (range: $0.00 \leq r \leq 30.00$; you can leave it blank for the default value of 1.80)

Catalogs: MAGIC 2FGL 3FGL (you may choose more than one)

[2] Cross check of the MAGIC catalog and the Fermi2 catalog

[3] Cross check of the MAGIC catalog and the Fermi3 catalog

This page was last modified on 1 September 2020

CONTACT:
Dijana Dominis Prester, University of Rijeka, dijana.dominis@gmail.com
Dario Hrupec, University of Osijek, dhrupec@gmail.com

© 2015-2020 D. Dominis Prester, D. Hrupec, J.-K. Zeljeznjak, Croatian MAGIC Consortium

Slika 2. - Glavna stranica mrežne aplikacije MAGIC Legacy









Na slici 2. vidimo glavnu stranicu mrežne aplikacije MAGIC Legacy. Aplikacija pretražuje dio neba obuhvaćen kružnicom polumjera r oko upisanih koordinata te ima ponuđeno više kataloga iz kojih može pretražiti podatke za taj dio neba. Katalogi koji su ponuđeni su: MAGIC, 2FGL i 3FGL. To su katalogi visoko-energijskih izvora gama-zraka opaženih pomoću teleskopa *Fermi*. Drugi katalog (2FGL) sadrži 1873 izvora u rasponu energija od 100 MeV do 100 GeV. Treći katalog (3FGL) sadrži gotovo dvostruko više podataka od drugog, 3033 izvora zajedno s njihovim lokacijama, spektralnim karakteristikama i mjesečnim svjetlosnim krivuljama. [10]

Jedini mogući unos podataka je u ICRS obliku koordinata. Vidimo sa slike da aplikacija očekuje podatke za rektascenziju i deklinaciju u satnom obliku. Cilj naše nadogradnje je omogućiti osim ICRS oblika i galaktički oblik koordinata te

osim satnog oblika omogućiti i upis koordinata u radijanima, stupnjevima i decimalnim brojevima i dodati četvrti *Fermi* katalog na popis kataloga iz kojih pretražujemo podatke.

7. Datoteke potrebne za rad aplikacije

Na slici 3. se nalaze sve datoteke koje su potrebne za rad aplikacije.

Name	Date modified	Type	Size
 checkMF2.php	28.6.2015. 18:32	PHP File	1 KB
 checkMF3.php	28.6.2015. 18:32	PHP File	1 KB
 fermi2.txt	29.4.2015. 11:29	Text Document	65 KB
 fermi3.txt	29.4.2015. 11:29	Text Document	105 KB
 find.py	1.9.2020. 13:04	PY File	9 KB
 index.html	1.9.2020. 12:35	Chrome HTML Do...	4 KB
 logo.jpg	30.4.2015. 10:44	JPG File	106 KB
 magic.txt	3.9.2020. 13:20	Text Document	37 KB

Slika 3. - Datoteke koje aplikacija koristi

Datoteke checkMF2.php i checkMF3.php se koriste za „crosscheck“ podataka. Datoteke fermi2.txt, fermi3.txt i magic.txt su katalozi iz kojih pretražujemo podatke. Katalozi su zapravo baze podataka koje se nalaze u tekstualnim datotekama aplikacije. Na serveru se nalaze skripte koje ažuriraju magic.txt bazu podataka međutim za ovaj rad one nam nisu potrebne. Datoteka find.py je skripta napisana u Pythonu koja odrađuje glavni dio posla. Datoteka index.html je datoteka koja prikazuje glavnu stranicu aplikacije te se pomoću nje prebacuju ulazni parametri u Python skriptu potrebni za pretraživanje podataka. Također služi za stiliziranje stranice u smislu određivanja položaja elemenata stranice. Datoteka logo.jpg je logo MAGIC Legacy aplikacije.

Datoteke koje sadrže podatke iz *Fermi* kataloga, sadrže samo tri stupca. Prvi stupac je naziv kataloga, drugi stupac naziv objekta, preostala dva stupca su rekstacenzija i deklinacija (koordinate objekta).

Datoteka magic.txt sadrži sedam stupaca podataka. Prvi je stupac naziv objekta, drugi stupac su podaci za „WOBBLE“ pozicije. Treći je stupac interval vremena u kojem se snimao objekt dok na nebu nije bilo mjeseca. Četvrti stupac je interval vremena u kojem se snimao objekt dok je na nebu bio mjesec. Peti i šesti stupac su koordinate objekta; rektascenzija u satnom obliku i deklinacija u stupnjevima. Zadnji stupac su podaci o periodima snimanja teleskopa za koje je objekt sniman.

8. Dio aplikacije napisan u programskom jeziku Python

Aplikacija s glavne stranice putem HTML koda poziva skriptu naziva find.py. Skripta pretražuje podatke iz gore navedenih kataloga te ih ispisuje nazad formatirane pomoću HTML oznaka kako bi ih internet preglednik na odgovarajući način mogao ispisati.

Glavni dio skripte je funkcija „calculate“. Funkcija za ulazne parametre uzima dvije koordinate (rektascenziju i deklinaciju u decimalnom obliku; rektascenzija u satnom obliku a deklinacija u stupnjevima), polumjer kružnice i nazive kataloga koje želimo pretražiti. Funkcija tada provjerava redom sve objekte u označenim katalogima i njihove udaljenosti od naše koordinate. Ako je udaljenost manja ili jednaka polumjeru kružnice koji smo naveli objekt i njegovi podaci će se ispisati u rezultatima. [12]

Skripta također koristi CGI (Common Gateway Interface) Python modul. To je modul koji se koristi za razmjenu podataka između web servera i skripte. Pomoću tog modula povlače se podaci iz index.html datoteke u find.py skriptu.

Podaci koji su nam bitni su koordinate i katalogi koje pretražujemo. Problem aplikacije je što podatke za koordinate prima samo u jednom obliku. Rektascenzija je u satnom obliku gdje su sati, minute i sekunde odvojeni dvotočkom (hh:mm:ss). Deklinacija je u stupnjevima, ali su stupnjevi, minute i

sekunde također odvojeni dvotočkom (dd:mm:ss). Aplikacija u datoteci index.html očekuje svaki broj zasebno; za sate, minute i sekunde te stupnjeve, minute i sekunde sprema kao odvojenu varijablu te ih tako šalje Python skripti. To nam ukupno daje šest varijabli. Python skripta tada ta tri broja za rektascenziju i tri broja za deklinaciju spaja u jedan, u decimalnom obliku za rektascenziju i jedan u decimalnom obliku za deklinaciju. Također mora uzimati u obzir predznak deklinacije. To zahtjeva uvođenje dodatne varijable koja je nazvana „minussign“.

```
if flag=="true":
    ra=ra_hh+ra_mm/60.+ra_ss/3600.
    if dec_dd>0:
        dec=dec_dd+dec_mm/60.+dec_ss/3600.
    if dec_dd<0:
        dec=dec_dd-dec_mm/60.-dec_ss/3600.
    if dec_dd==0 and minussign=="false":
        dec=dec_dd+dec_mm/60.+dec_ss/3600.
    if dec_dd==0 and minussign=="true":
        dec=dec_dd-dec_mm/60.-dec_ss/3600.
```

Slika 4. - Dio Python skripte koji transformira upisane varijable u decimalni oblik

Slika 5. prikazuje dio skripte koji ulazne varijable pretvara u dvije izlazne koje tada u decimalnom obliku koristi za gore navedenu „calculate“ funkciju. Sa slike 5. možemo zaključiti da funkcija „calculate“ očekuje rektascenziju u satnom decimalnom obliku, a deklinaciju u decimalnim stupnjevima. Kako bi izbjegli dodatno kompliciranje i mijenjanje same „calculate“ funkcije, u nadogradnji ćemo sve dodatne oblike koordinata svesti na ova dva oblika (satni decimalni i decimalni stupnjevi), pa ih onda ubaciti u „calculate“ skriptu.

9. Implementacija četvrtog *Fermi*-kataloga

Rad na nadogradnji aplikacije počinjemo s implementacijom četvrtog *Fermi* kataloga. Drugi i treći katalozi su već ubačeni u aplikaciju te se njihove baze podataka nalaze u ranije opisanim datotekama.

Teleskop *Fermi* skenira cijelo nebo svaka tri sata. Teleskop detektira gama-zrake u rasponu energija od 20 MeV do 300 GeV. S obzirom na to da s porastom energija količina čestica koje izvor emitira eksponencijalno opada, visoko energijske gama-zrake su manje učestale nego niskoenergijske gama-zrake. Iznad 10 GeV za neke izvore *Fermi* opazi u prosjeku tek jednu gama-zraku svakih četiri mjeseca. *Fermi* posjeduje vidno polje koje zatvara kut od 60° dok MAGIC teleskopi posjeduju vidno polje koje zatvara kut od $3,6^\circ$. Teleskopi koji se nalaze na Zemljinoj površini također ne mogu snimati po danu, po lošem vremenu i za punog mjeseca. MAGIC teleskopi i općenito teleskopi koji se nalaze na Zemljinoj površini ipak imaju jednu prednost; imaju puno veću efektivnu površinu detektora te zbog površine Čerenkovljevog bazena (površinu na koju na tlu padaju Čerenkovljevi fotoni iz jednog pljuska čestica u atmosferi iniciranog upadom jedne gama-zrake), su bolji za promatranje izvora visokih energija. [12]

Četvrti *Fermi* katalog rezultat je opažanja izvora gama-zraka tijekom prvih osam godina promatranja neba. 4FGL-DR2 je nadogradnja četvrtog kataloga (4FGL) koju smo mi zapravo implementirali u aplikaciju. Nadograđeni katalog je rezultat opažanja koje je trajalo deset godina, znači dvije godine više nego prvobitni četvrti katalog. Točno razdoblje opažanja na koje se katalog odnosi je od 4. 8. 2008 do 2. 8. 2018. Energijski raspon kojeg katalog pokriva je od 50 MeV do 1 TeV. Katalog sadrži 50-ak parametara za svaki identificiran izvor. Za našu aplikaciju su bitna tri parametra, a to su naziv objekta, rektascenzija i deklinacija. [16]

Prvi korak bio je preuzeti katalog sa službene *Fermi* stranice. [13] Katalog naravno posjeduje puno više podataka od onoga što je potrebno za aplikaciju. Katalog je spremljen u .fit datoteku. FITS (Flexible Image Transport System) je tip datoteke koji se najčešće koristi u astronomiji. Osim pakiranja velikog broja slikovnih datoteka može se koristiti i kao baza podataka sa tablicama koje ne sadrže slikovne datoteke. Alati s kojima možemo pristupiti podacima iz .fit

datoteke su mnogobrojni, od programskih jezika do posebnih programa namjenjenih za .fit datoteke.

Program kojim se pristupilo podacima u .fit datoteci je *fv*. [14] Taj program je dio šireg spektra programa koje upotrebljava NASA (National Aeronautics and Space Administration). Nakon što se pristupi .fit datoteci na ekranu se dobiju svi podatci koji se nalaze u katalogu. Program ima mogućnost odabira stupaca koje se tada može prebaciti u tekstualnu datoteku. Nakon što se naziv kataloga, naziv objekta, rektascenzija i deklinacija prebace u zasebnu tekstualnu datoteku, potrebno je tu tekstualnu datoteku malo urediti. Program koji se koristi za to je Notepad++. [15] Notepad++ ima mogućnosti brisanja cijelih stupaca i redova naše datoteke. Nakon što maknemo podatke koji nam ne trebaju iz tekstualne datoteke; na primjer nazive stupaca, tekstualnu datoteku možemo koristiti kao bazu podataka za našu aplikaciju.

4FGL J0000.3-7355	0.0983	-73.9220
4FGL J0000.5+0743	0.1375	7.7273
4FGL J0001.2+4741	0.3126	47.6859
4FGL J0001.2-0747	0.3151	-7.7971
4FGL J0001.5+2113	0.3815	21.2183
4FGL J0001.6-4156	0.4165	-41.9425
4FGL J0002.1-6728	0.5378	-67.4746
4FGL J0002.1+6721c	0.5419	67.3578
4FGL J0002.3-0815	0.5937	-8.2652
4FGL J0002.4-5156	0.6131	-51.9355
4FGL J0002.7+7220	0.6786	72.3479

Slika 5. - Primjer nekoliko objekata u bazi podataka Fermi 4 kataloga

Slika 6. prikazuje način na koji izgleda naša baza podataka nakon što smo obrisali nepotrebne podatke. Prvi stupac je naziv kataloga, drugi stupac je naziv objekta, treći stupac je rektascenzija u stupnjevima, a četvrti stupac je deklinacija u stupnjevima.

Nakon što smo dobili tekstualnu bazu podataka potrebno je omogućiti korištenje te baze podataka u aplikaciji. U index.html datoteku dodajemo još jedan potvrdni

okvir uz već postojeća tri. Sada imamo po jedan potvrdni okvir za magic.txt, fermi2.txt, fermi3.txt i fermi4.txt baze podataka. U ovisnosti o odabranim potvrdnim okvirima aplikacija će iz navedenih baza podataka pretraživati podatke. Nakon što smo dodali potvrdni okvir potrebno je u python skripti dodati dio koda koji će pretraživati podatke iz nove fermi4.txt baze podataka.

```
if isF4 == "ON":
    fermi = open("fermi4.txt", "r")
    print("<b>Sources in the FERMI4 catalog:</b>")
    print("<table border=1 style='width:50%'>")
    name_F = []
    declination_F = []
    rasc_F = []
    for line in fermi:
        if len(line.strip()) != 0 and "#" not in line:
            words = line.split()
            name_F.append(words[0] + " " + words[1])
            rasc_F.append(float(words[2]))
            declination_F.append(float(words[3]))
    fermi.close()
    count_single = 0
    for i in range(len(name_F)):
        output = ""
        r0 = distance(rasc_F[i], declination_F[i], ra, dec)
        if r0 <= r_input:
            count_single += 1
            rasca = antitransform(rasc_F[i] / 15)
            output = "<td>" + name_F[i] + "</td><td>" + "{:.3f}".format(
                r0) + "</td><td>" + rasca + "</td><td>" + antitransform(declination_F[i]) + "</td>"
            print("<tr>" + output + "</tr>")
    if count_single == 0:
        print("<br>")
        print("No sources found in the FERMI4 catalog")
        print("<br>")
    print("</table>")
    print("<br>")
```

Slika 6. - Dio koda koji pretražuje podatke iz FERMI 4 kataloga

Kao što vidimo na slici 7., kod je sličan kodu za preostale dvije *Fermi* baze podataka. Potrebno je dodati provjeru za dodatni potvrdni okvir i staviti drugi naziv datoteke u kojoj se nalazi baza podataka (u ovom slučaju fermi4.txt). Nakon što se provjeri koji objekti zadovoljavaju tražene uvjete, ispisuje ih se u internet pretraživač.

10. Implementacija mogućnosti unosa koordinata na različite načine

Cilj ovog dijela nadogradnje je omogućiti korisniku aplikacije unos koordinata objekta na više od jednog načina. Jedini mogući način prije nadogradnje bio je nepraktičan. Rektascenzija se upisivala u satnom obliku gdje su sati, minute i sekunde imale zasebne varijable. Deklinacija se upisivala u stupnjevima gdje su stupnjevi, minute i sekunde također imali zasebne varijable. Osim samih koordinata cilj je dodati i još jedan astronomski okvir. Aplikacija prije nadogradnja podržava samo ICRS, a poslije nadogradnje podržava ICRS i galaktički okvir, to jest oblik koordinata.

Nadogradnja omogućava korisniku odabir okvira u sklopu kojeg želi upisati koordinate. Ako ne odabere okvir ne može ni pristupiti upisu koordinata. Nakon što odabere okvir otvara mu se dio aplikacije gdje upisuje rektascenziju i deklinaciju za ICRS okvir te dužinu i širinu za galaktički okvir. Mnogobrojne varijable koje su se koristile prije nadogradnje su u nadogradnji zamjenjene sa jednom za rektascenziju i jednom za deklinaciju te simetrično tome jednom za dužinu i jednom za širinu.

Svaka od koordinata koja se upisuje, zahvaljujući klasi SkyCoord, može biti upisana na više načina. Korisnik odabire oblik kuta u kojem želi upisati koordinatu. Mogući oblici su: radijani, stupnjevi i satni oblik. Radijane je moguće upisati u decimalnom obliku. Stupnjeve se može upisati u decimalnom obliku, u obliku gdje je separator između stupnjeva, minuta i sekunda razmak, u obliku gdje su separatori redom slova „d“ „m“ „s“ (stupnjevi, minute i sekunde) i oblik gdje su separatori „°“ „‘“ i „““. Satni oblik također podržava decimalni oblik, oblik gdje je separator između sati, minuta i sekunda razmak, te oblik gdje su separatori redom slova „h“ „m“ „s“ (sati, minute i sekunde).

Različiti načini unosa koordinata su mogući zahvaljujući fleksibilnosti klase SkyCoord.

```
ra = Angle(ra, unit=angletype_ra)
dec = Angle(dec, unit=angletype_dec)

obj = SkyCoord(ra, dec, unit=(angletype_ra, angletype_dec), frame=cframe)

if cframe == 'icrs':
    ra = obj.ra.to_string(unit=u.hourangle, decimal=True)
    dec = obj.dec.to_string(unit=u.deg, decimal=True)
elif cframe == 'galactic':
    obj = obj.transform_to('icrs')
    ra = obj.ra.to_string(unit=u.hourangle, decimal=True)
    dec = obj.dec.to_string(unit=u.deg, decimal=True)
else:
    flag = "false"

ra = ra.astype(float)
dec = dec.astype(float)

r = float(r)
```

Slika 7. - Glavni dio koda koji omogućuje unos različitih oblika koordinata i okvira

Slika 8. nam pokazuje glavni dio koda koji omogućava implementaciju nadogradnje. Skripta iz index.html datoteke povlači varijable ra i dec što odgovara rektascenziji i deklinaciji, te oblik okvira i tipove kuteva koji su upisani. S obzirom da su varijable za galaktički okvir dužina i širina logično je zaključiti da ne bi imalo smisla ubacivati rektascenziju i deklinaciju. Zbog jednostavnosti nisu se uvodile još dvije varijable za dužinu i širinu nego se rektascenzija, ako je to potrebno, koristi kao dužina, a deklinacija kao širina. Sam naziv varijable nije toliko važan.

Nakon što povuče vrijednosti varijabli za rektascenziju i deklinaciju prvo stvara objekt klase Angle. Angle je klasa koja je kao i SkyCoord dio Astropy paketa. Omogućava nam lakše korištenje SkyCoord objekta jer Angle objekt možemo koristiti kao početnu varijablu kod instanciranja SkyCoord objekta. Angle objekt

za instanciranje traži iznos kuta i tip kuta (stupnjevi, radijani itd.). Tip kuta se definira pomoću alata iz paketa `astropy.units`. U tom paketu postoji mogućnost definiranja varijable kao točno određenog tipa kuta (radijani, stupnjevi itd.). Skripta iznos i tip kuta povlači iz `index.html` datoteke.

Pomoću instancirana dva kuta sada se stvara `SkyCoord` objekt. `SkyCoord` klasa za instanciranje objekta traži dva kuta, oblik kuta u kojem su kutevi upisani, te okvir koji se koristi. Nakon instanciranja objekta; ako je okvir ICRS pomoću metode `to_string` prebacujemo rektascenziju i deklinaciju u decimalni oblik. Ovdje se koristi decimalni oblik zato što glavna funkcija `calculate` koristi rektascenziju i deklinaciju u decimalnom obliku.

Ako se radi o galaktičkom okviru prvo se metodom `transform_to` `SkyCoord` objekt prebaci u ICRS okvir. Nakon toga opet metodom `to_string` rektascenziju i deklinaciju se prebaci u decimalni oblik. Ovakav način rješavanja problema omogućuje da se neovisno o tipu ulaznog kuta i okvira, rektascenzija i deklinacija u glavnu funkciju ubacuju u decimalnom ICRS obliku.

Problem nastaje jer metoda `to_string` iako prebaci rektascenziju i deklinaciju u decimalni oblik, varijable za rektascenziju i deklinaciju postaju tekstualne varijable. Funkcija `calculate` za rektascenziju i deklinaciju očekuje numeričke varijable. U `numpy` paketu postoji metoda `.astype()` koja može određenu varijablu prebaciti u željeni oblik. U kodu su rektascenzija i deklinacija pomoću te metode prebačene u decimalni numerički oblik i postaju spremne za daljnu obradu u funkciji `calculate`.

Zbog ovakvog načina obrade varijabli prije unosa u `calculate` funkciju ne mora se izmijeniti `calculate` funkcija. Time je zadatak nadogradnje pojednostavljen.

11. Modifikacija index.html datoteke i dodavanje javascript skripte

Index.html datoteka određuje izgled aplikacije i prosljeđuje nazive i vrijednosti varijabli Python skripti. Prva promjena je bila definiranje paragrafa unutar html koda. Prije nadogradnje html kod nije sadržavao ni kontejnere ni paragrafe. Paragrafe se definira jer nam omogućuju dodjeljivanje klasa određenim paragrafima koje se kasnije pomoću javascript skripte sakriju ili pokažu u internet pregledniku.

Javascript je skriptni programski jezik koji se izvršava u pregledniku. Javascript se upotrebljava zbog paketa jQuery. jQuery sadrži metode i funkcije napisane u javascript programskom jeziku. Te metode i funkcije olakšavaju manipulaciju sa html objektima, u ovom slučaju paragrafima i kontenjerima.

```
$(document).ready(function() {
  $(".icrstxt").hide()
});
$(document).click(function() {
  if($('.icrsbtn').is(':checked')) { $('.icrstxt').show(500); $('.galactictxt').hide() }
});

$(document).ready(function() {
  $(".galactictxt").hide()
});
$(document).click(function() {
  if($('.galacticbtn').is(':checked')) { $('.galactictxt').show(500); $('.icrstxt').hide() }
});
```

Slika 8. - Javascript skripta naziva main.js

Slika 9. nam prikazuje javascript dio koda. Taj dio koda se nalazi u mapi scripts u datoteci main.js. Mapa scripts i datoteka main.js su potpuno novi dijelovi aplikacije. Pomoću koda na slici 9. sakrivamo i prikazujemo paragrafe u odnosu na okvir koji korisnik izabere. Ako korisnik izabere ICRS okvir pokaže se paragraf gdje se unosi rektascenzija i deklinacija, a ako korisnik izabere galaktički

okvir pokaže se paragraf gdje se unosi dužina i širina. Cilj korištenja ove skripte je smanjiti „nered“ na stranici te je učiniti preglednijom.

U <head> dijelu html koda se također dodalo par linija koda koji se odnose na CSS (Cascading Style Sheets). CSS je stilski jezik koji nam omogućuje stiliziranje html objekata. Pomoću CSS-a se definiralo odmake od rubova kontejnera i paragrafa te se dodalo crnu horizontalnu crtu ispod svakog paragrafa i kontejnera. Crta se dodala pomoću CSS-a da zamjeni pomalo zastarjeli način ispisivanja horizontalne crte pomoću html koda.

Osim samog izgleda aplikacije promjena je i kod unosa varijabli. Više nije potrebno pisati zasebno npr. stupnjeve minute i sekunde nego se to sada sve odrađuje u jednom tekstualnom okviru. Osim već navedenih potvrdnih okvira za ICRS i galaktički okvir, kod svakog unosa kuta dodani su potvrdni okviri za tip kuta koji se unosi. Uvjet za uspješno dobivanje podataka iz aplikacije je definirati pomoću potvrdnog okvira osim samog koordinatnog okvira i tip kuta koji se upisuje. Naravno ako se u tekstualni okvir upiše nešto što nema smisla ili što SkyCoord objekt ne prepoznaje, aplikacija izbacuje grešku.

12. Završetak nadogradnje

Nakon implementacije svih gore navedenih nadogradnji, aplikacija poprima završni izgled.



[1] Search in a radius (r) around a position on the celestial sphere

Choose frame: ICRS GALACTIC

Angle type for RA: Radian Degrees Hourangle

Right ascension (VALID INPUTS: rad \rightarrow x.xx | degrees \rightarrow xxdxxmxxx or xx xx xx or xx°xx'xx" or x.xx | hourangle \rightarrow xxhxxmxxx or xx xx xx or x.xx)

Angle type for DEC: Radian Degrees Hourangle

Declination (VALID INPUTS: rad \rightarrow x.xx | degrees \rightarrow xxdxxmxxx or xx xx xx or xx°xx'xx" or x.xx | hourangle \rightarrow xxhxxmxxx or xx xx xx or x.xx)

Radius of interest (r) range: 0.00 $\leq r \leq$ 30.00; you can leave it blank for the default value of 1.80

Catalogs: MAGIC 2FGL 3FGL 4FGL (you may choose more than one)

[2] Cross check of the MAGIC catalog and the Fermi2 catalog

[3] Cross check of the MAGIC catalog and the Fermi3 catalog

This page was last modified on 1 September 2020

CONTACT:
Dijana Dominis Prester, University of Rijeka, dijana.dominis@gmail.com

Slika 9. - Glavna stranica aplikacije MAGIC Legacy nakon nadogradnje

Na slici 10. se vidi izgled aplikacije nakon implementacije nadogradnji. U ovom slučaju odabran je ICRS okvir. Korisnik upisuje rektascenziju i deklinaciju te odabire tip kuta u kojem ih upisuje. Kutevi se mogu upisati na već opisane načine. Osim samog izbora okvira također ima mogućnosti izabrati pretragu podataka iz posljednjeg četvrtog *Fermi* kataloga. Aplikacija je nakon nadogradnje zadržala svoj jednostavni izgled.

13. Buduće nadogradnje

Aplikacija i dalje ima određene nedostatke. Jedan njih je nemogućnost aplikacije da točno detektira što je netočno kod unosa. Ako korisnik upiše nešto što nema smisla u tekstualni okvir gdje se očekuje koordinata, aplikacija će izbaciti grešku, ali netko tko ne zna točan način na koji aplikacija funkcionira neće moći lako razumijeti iz kojeg točno razloga se javlja greška. HTML dio aplikacije je i dalje malo zastario. Uobičajene prakse su se promjenile te tu postoji prostora za nadogradnju.

Osim izgleda i funkcionalosti stranice moguće je unaprijediti sigurnost aplikacije. Iako se prije pristupanja mrežnoj stranici, na kojoj se nalazi aplikacija, treba

ulogirati sa korisničkim imenom i lozinkom, aplikacija u samom kodu nema nikakve mjere sigurnosti.

14. Zaključak

Cilj diplomskog rada je bio nadograditi aplikaciju MAGIC Legacy kako bi olakšali njezinu upotrebu. Implementacija nadogradnje je odrađena upotrebom objekata i metoda iz Python paketa Astropy. Omogućeni su izbori okvira i različiti načini upisa koordinata te je pomoću javascripta aplikacija učinjena preglednijom. Dodana je mogućnost pretraživanja podataka iz četvrtog *Fermi* kataloga. HTML dio aplikacije je učinjen preglednijim. Postoje mogućnosti za daljnu nadogradnje aplikacije.

15. Literatura

[1] DIAMANTIS, Aristidis, et al. The contribution of Maria Sklodowska-Curie and Pierre Curie to Nuclear and Medical Physics. A hundred and ten years after the discovery of radium. *Hell J Nucl Med*, 2008, 11.1: 33-38.

[2] DI SCIASCIO, Giuseppe. Ground-based Gamma-Ray Astronomy: an Introduction. In: *Journal of Physics: Conference Series*. IOP Publishing, 2019. p. 012003.

[3] Introduction, Official MAGIC page. Semptember 2020. URL <https://magic.mpp.mpg.de/newcomers/introduction0/>

[4] Cherenkov Telescopes, Official MAGIC page. Semptember 2020. URL <https://magic.mpp.mpg.de/newcomers/cherenkov-telescopes/>

[5] MALING, Derek Hylton. *Coordinate systems and map projections*. Elsevier, 2013.

[6] Glactic coordinate, Encyclopaedia Britannica, Semptember 2020. URL: <https://www.britannica.com/science/galactic-coordinate>

[7] Sky & Telescope The Essential Guide to Astronomy page. Semptember 2020. URL <https://skyandtelescope.org/astronomy-resources/right-ascension-declination-celestial-coordinates/>

[8] ONG, Rene A. Very high-energy gamma-ray astronomy. *Physics Reports*, 1998, 305.3-4: 93-202.

[9] Official Astropy page. Semptember 2020. URL <https://www.astropy.org/about.html>

[10] ACERO, Fabio, et al. *Fermi* large area telescope third source catalog. *The Astrophysical Journal Supplement Series*, 2015, 218.2: 23.

- [11] Official SkyCoord page. Semptember 2020. URL <https://docs.astropy.org/en/stable/api/astropy.coordinates.SkyCoord.html>
- [12] ŽELJEZNJAK, Jelena-Kristina. *Sources emitting gamma-rays observed in the field of view of the MAGIC telescopes*. 2017. Odjel za fiziku, Sveučilište u Rijeci.
- [13] Official NASA *Fermi* page. Semptember 2020 URL https://fermi.gsfc.nasa.gov/ssc/data/access/lat/10yr_catalog/gll_psc_v24.fit
- [14] Official NASA fv tool page. Semptember 2020 https://heasarc.gsfc.nasa.gov/docs/software/ftools/fv/fv_download.html
- [15] Official Notepad++ page. Semptember 2020. URL <https://notepad-plus-plus.org/downloads/>
- [16] Official NASA *Fermi* page for *Fermi* fourth catalog URL https://fermi.gsfc.nasa.gov/ssc/data/access/lat/10yr_catalog/


```

        <br>
        <br>
        <input type="submit" value="SUBMIT [1]" />
    </p>
</form>
<div>
    <form action="checkMF2.php" method="post" target="_blank">
        <font size="3">[2] Cross check of the MAGIC catalog and the Fermi2
catalog</font>
        <input type="submit" value="SUBMIT [2]" />
    </form>
</div>
<div>
    <form action="checkMF3.php" method="post" target="_blank">
        <font size="3">[3] Cross check of the MAGIC catalog and the Fermi3
catalog</font>
        <input type="submit" value="SUBMIT [3]" />
    </form>
</div>

<font color="gray" size="2">
This page was last modified on 1 September 2020<br>
<br>
CONTACT:<br>
Dijana Dominis Prester, University of Rijeka,
<script language="JavaScript">
    var username = "dijana.dominis";
    var hostname = "gmail.com";
    var linktext = username + "@" + hostname ;
    document.write("<a href='" + "mail" + "to:" + username + "@" +
hostname + "'" + "style='color: #808080;'">" + linktext + "</a>");
</script><br>
Dario Hrupec, University of Osijek,
<script language="JavaScript">
    var username = "dhrupec";
    var hostname = "gmail.com";
    var linktext = username + "@" + hostname ;
    document.write("<a href='" + "mail" + "to:" + username + "@" +
hostname + "'" + "style='color: #808080;'">" + linktext + "</a>");
</script><br>
<br>
&copy; 2015-2020 D. Dominis Prester, D. Hrupec, J.-K. Źeljenzjak,
Croatian MAGIC Consortium
</font>
</body>
</html>

```

16.2 find.py datoteka

```

#!C:\Python38\python.exe

# jkzeljeznjak@gmail.com
# dario.hrupec@irb.hr
# 8th September 2015

import cgi
import cgitb
from math import *

```

```

import re
from astropy.coordinates import SkyCoord
from astropy import units as u
from astropy.coordinates import Angle

def isfloat(value):
    try:
        float(value)
        return True
    except ValueError:
        return False

def transform(x):
    num = x.replace("+", "")
    num = num.replace("-", "")
    num = re.split(':', num)
    if x[0] == "-":
        return -(int(num[0]) + int(num[1]) / 60. + float(num[2]) / 3600.)
    else:
        return (int(num[0]) + int(num[1]) / 60. + float(num[2]) / 3600.)

def antitransform(x):
    out = ""
    if x >= 0:
        out += "+"
    out += str(int(x)) + ":"
    x = abs(x) % 1 * 60
    out += str(int(x)) + ":"
    x = abs(x) % 1 * 60
    out += str(int(x))
    return out

def distance(ra1, de1, ra2, de2):
    return sqrt((ra1 - ra2)**2 + (de1 - de2)**2)

def trigonometry(k, angle, slope, ra1, de1, ra2, de2):
    if k % 2 == 0:
        x = angle * cos(slope)
        y = angle * sin(slope)
    else:
        x = -angle * cos(slope)
        y = -angle * sin(slope)
    return distance(ra1 + x, de1 + y, ra2, de2)

def calculate(ra, dec, r_input, isM, isF2, isF3, isF4):
    ra *= 15
    if isM == "ON":
        print("<b>Sources in the MAGIC catalog:</b>")
        print("<table border=1 style='''width:100%'''>")
        magic = open("magic.txt", "r")
        name_M = []
        declination_M = []
        rasc_M = []
        wobble = []
        time_no = []

```

```

time_moon = []
periods = []
for line in magic:
    if len(line.strip()) != 0 and "#" not in line:
        words = line.split()
        name_M.append(words[0])
        wobble.append(words[1])
        time_no.append(float(words[2]))
        time_moon.append(float(words[3]))
        rasc_M.append(words[4])
        declination_M.append(words[5])
        los = [int(per) for per in words[6:]]
        periods.append(los)
magic.close()
count_single = 0
for i in range(len(name_M)):
    output = ""
    r0 = distance(
        15 * transform(rasc_M[i]), transform(declination_M[i]), ra,
dec)
        if r0 <= r_input or (r_input == 1.8 and r0 <= r_input + 0.4):
            count_single += 1
            output = "<td>" + name_M[i] + "</td><td>" +
"{:.3f}".format(
                r0) + "</td><td>" + rasc_M[i] + "</td><td>" +
declination_M[i]
            output += "</td><td>" + "Dark_time(h)=" + str(
                time_no[i]) + "</td><td>" + "Time_Moon(h)=" +
str(time_moon[i]) + "</td>"
            output += "<td>" + "Periods: " + str(periods[i]) + "</td>"
            num = 0
            if len(wobble[i]) > 8:
                num = 4
                angle1 = float(wobble[i][1:4]) / 100
                angle2 = float(wobble[i][9:12]) / 100
                slope1 = int(wobble[i][5:8])
                slope2 = int(wobble[i][13:])
            elif len(wobble[i]) == 8:
                num = 2
                angle1 = float(wobble[i][1:4]) / 100
                slope1 = int(wobble[i][5:8])
            count = 0
            for k in range(num):
                if k < 2:
                    slope = slope1
                    angle = angle1
                else:
                    slope = slope2
                    angle = angle2
                r = trigonometry(
                    k, angle, slope, 15 * transform(rasc_M[i]),
transform(declination_M[i]), ra, dec)
                if r <= r_input:
                    count += 1
            if wobble[i] == "None" and r_input == 1.8:
                output += "<td>" + "This MAGIC has no wobbles" +
"</td>"
            elif r_input == 1.8:
                output += "<td>" + \
                    str(count) + " out of " + \
                    str(num) + " wobbles" + "</td>"

```

```

        count += 1
        print("<tr>" + output + "</tr>")
if count_single == 0:
    print("<br>")
    print("No sources found in the MAGIC catalog")
    print("<br>")
print("</table>")
print("<br>")
print("<br>")

if isF2 == "ON":
    fermi = open("fermi2.txt", "r")
    print("<b>Sources in the FERMI2 catalog:</b>")
    print("<table border=1 style='''width:50%'''>")
    name_F = []
    declination_F = []
    rasc_F = []
    for line in fermi:
        if len(line.strip()) != 0 and "#" not in line:
            words = line.split()
            name_F.append(words[0] + " " + words[1])
            rasc_F.append(float(words[2]))
            declination_F.append(float(words[3]))
    fermi.close()
    count_single = 0
    for i in range(len(name_F)):
        output = ""
        r0 = distance(rasc_F[i], declination_F[i], ra, dec)
        if r0 <= r_input:
            count_single += 1
            rasca = antitransform(rasc_F[i] / 15)
            output = "<td>" + name_F[i] + "</td><td>" +
"{:.3f}".format(
                r0) + "</td><td>" + rasca + "</td><td>" +
antitransform(declination_F[i]) + "</td>"
            print("<tr>" + output + "</tr>")
    if count_single == 0:
        print("<br>")
        print("No sources found in the FERMI2 catalog")
        print("<br>")
    print("</table>")
    print("<br>")
    print("<br>")

if isF3 == "ON":
    fermi = open("fermi3.txt", "r")
    print("<b>Sources in the FERMI3 catalog:</b>")
    print("<table border=1 style='''width:50%'''>")
    name_F = []
    declination_F = []
    rasc_F = []
    for line in fermi:
        if len(line.strip()) != 0 and "#" not in line:
            words = line.split()
            name_F.append(words[0] + " " + words[1])
            rasc_F.append(float(words[2]))
            declination_F.append(float(words[3]))
    fermi.close()
    count_single = 0
    for i in range(len(name_F)):
        output = ""

```

```

        r0 = distance(rasc_F[i], declination_F[i], ra, dec)
        if r0 <= r_input:
            count_single += 1
            rasca = antitransform(rasc_F[i] / 15)
            output = "<td>" + name_F[i] + "</td><td>" +
"{:.3f}".format(
                r0) + "</td><td>" + rasca + "</td><td>" +
antitransform(declination_F[i]) + "</td>"
            print("<tr>" + output + "</tr>")
        if count_single == 0:
            print("<br>")
            print("No sources found in the FERMI3 catalog")
            print("<br>")
        print("</table>")
        print("<br>")

```

```

if isF4 == "ON":
    fermi = open("fermi4.txt", "r")
    print("<b>Sources in the FERMI4 catalog:</b>")
    print("<table border=1 style='width:50%'>")
    name_F = []
    declination_F = []
    rasc_F = []
    for line in fermi:
        if len(line.strip()) != 0 and "#" not in line:
            words = line.split()
            name_F.append(words[0] + " " + words[1])
            rasc_F.append(float(words[2]))
            declination_F.append(float(words[3]))
    fermi.close()
    count_single = 0
    for i in range(len(name_F)):
        output = ""
        r0 = distance(rasc_F[i], declination_F[i], ra, dec)
        if r0 <= r_input:
            count_single += 1
            rasca = antitransform(rasc_F[i] / 15)
            output = "<td>" + name_F[i] + "</td><td>" +
"{:.3f}".format(
                r0) + "</td><td>" + rasca + "</td><td>" +
antitransform(declination_F[i]) + "</td>"
            print("<tr>" + output + "</tr>")
        if count_single == 0:
            print("<br>")
            print("No sources found in the FERMI4 catalog")
            print("<br>")
        print("</table>")
        print("<br>")

```

```

cgitb.enable()
form = cgi.FieldStorage()

```

```

ra = form.getvalue('ra')
dec = form.getvalue('dec')
r = form.getvalue('r')

```

```

flag = "true"

```

```

if r == None:

```

```

r = 1.8

if form.getvalue('angletype') == 'deg':
    angletype_ra = u.deg
elif form.getvalue('angletype') == 'rad':
    angletype_ra = u.rad
elif form.getvalue('angletype') == 'hourangle':
    angletype_ra = u.hourangle
else:
    flag = "false"

if form.getvalue('angletype2') == 'degdec':
    angletype_dec = u.deg
elif form.getvalue('angletype2') == 'raddec':
    angletype_dec = u.rad
elif form.getvalue('angletype2') == 'hourangledec':
    angletype_dec = u.hourangle
else:
    flag = "false"

if form.getvalue('frame') == 'icrs':
    cframe = 'icrs'
elif form.getvalue('frame') == 'galactic':
    cframe = 'galactic'
else:
    flag = "false"

if form.getvalue('magic'):
    magic = "ON"
else:
    magic = "OFF"

if form.getvalue('fgl2'):
    fgl2 = "ON"
else:
    fgl2 = "OFF"

if form.getvalue('fgl3'):
    fgl3 = "ON"
else:
    fgl3 = "OFF"

if form.getvalue('fgl4'):
    fgl4 = "ON"
else:
    fgl4 = "OFF"

print("Content-type:text/html\r\n\r\n")
print("<html>")
print("<head>")
print("<title>Results</title>")
print("</head>")
print("<body>")

ra = Angle(ra, unit=angletype_ra)
dec = Angle(dec, unit=angletype_dec)

obj = SkyCoord(ra, dec, unit=(angletype_ra, angletype_dec), frame=cframe)

if cframe == 'icrs':

```

```

    ra = obj.ra.to_string(unit=u.hourangle, decimal=True)
    dec = obj.dec.to_string(unit=u.deg, decimal=True)
elif cframe == 'galactic':
    obj = obj.transform_to('icrs')
    ra = obj.ra.to_string(unit=u.hourangle, decimal=True)
    dec = obj.dec.to_string(unit=u.deg, decimal=True)
else:
    flag = "false"

ra = ra.astype(float)
dec = dec.astype(float)

r = float(r)

if flag == "false":
    print("<br><br>")
    print("<h2>Your input is wrong. Please, go back to the input
page.</h2>")
else:
    print("<h3>RA = %2.4f h [ICRS]</h3>" % (ra))
    print("<h3>DEC = %2.4f ° [ICRS]</h3>" % (dec))
    print("<h3>r = %2.2f</h3>" % (r))

print("<hr>")

if flag == "true":
    calculate(ra, dec, r, magic, fgl2, fgl3, fgl4)

print("</body>")
print("</html>")

```

16.3 main.js datoteka

```

$(document).ready(function() {
    $(".icrstxt").hide()
});
$(document).click(function() {
    if($('.icrsbtn').is(':checked')) { $('.icrstxt').show(500);
$('.galactictxt').hide() }
});

$(document).ready(function() {
    $(".galactictxt").hide()
});
$(document).click(function() {
    if($('.galacticbtn').is(':checked')) { $('.galactictxt').show(500);
$('.icrstxt').hide() }
});

```