

BATMAN - EU projekt za Acne Inversa

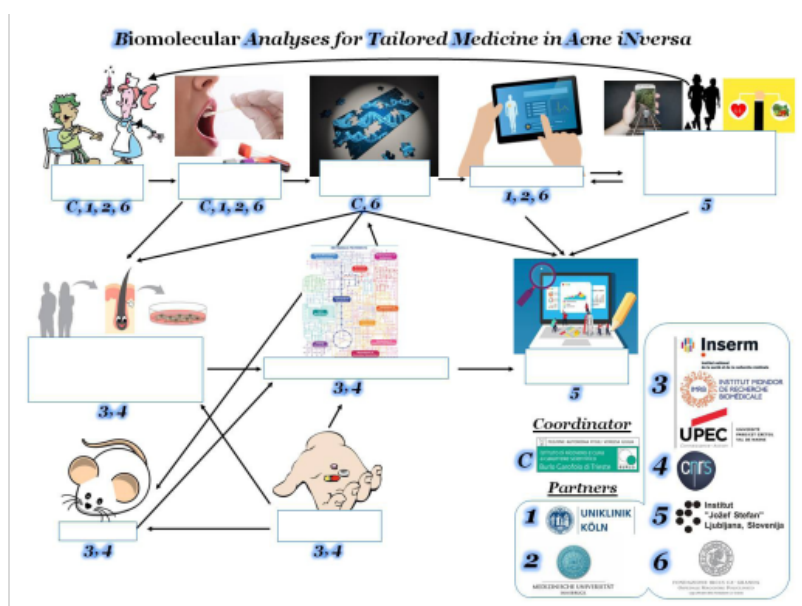
BATMAN - EU project for Acne Inversa

Zdenko Vuk
zdenko.vuk@ijs.si
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia

Anton Gradišek
anton.gradisek@ijs.si
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia

Jani Bizjak
jani.bizjak@ijs.si
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia

Matjaž Gams
matjaz.gams@ijs.si
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia



Slika 1: BATMAN skica

POVZETEK

V tem članku je predstavljen projekt BATMAN - Biomolecular Analyses for Tailored Medicine in Acne iNversa (Analiza biomarkerjev za personalizirano medicino acne inversa). Gre se za klinično-epidemiološko opazovalno študijo. Poleg tega je prikazana zasnova strežnika. Poudarek članka je na izvedbi spletne strani.

KLJUČNE BESEDE

zdravstvo, zbiranje podatkov, spletna aplikacija, Vue.js, Nuxt.js, Vuetify, single page application

ABSTRACT

This paper presents the project BATMAN - Biomolecular Analyses for Tailored Medicine in Acne iNversa (Analysis of biomarkers for personalized medicine acne inversa). It is a clinical-epidemiological observational study. Also, the server design is displayed. The focus of this paper is the implementation of the website.

KEYWORDS

healthcare, data collection, web app, Vue.js, Nuxt.js, Vuetify, single page application

1 UVOD

1.1 Opis problema

Acne Inversa (AI) je kronična vnetna bolezen, ki vključuje lasne mešičke, ki pacientom povzročajo veliko fizično in psihološko breme z velikimi stroški za zdravstvene sisteme[1][7]. Ponavadi se pojavi po puberteti in v Evropi prizadane 4% populacije[6]. Genetske različice, ki vplivajo na različne poti, povzročajo širok spekter fenotipov AI, sledenje genskim različicam pa je bistvene pomena za oblikovanje prilagojenih načinov zdravljenja.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

Information Society 2020, 5–9 October, 2020, Ljubljana, Slovenia

© 2020 Copyright held by the owner/author(s).

Cilj projekta je združiti medicinske, genetske, eksperimentalne podatke in podatke o življenjskem slogu, da bi ustvarili celostne zdravstvene evidence (HHR), ki nam bodo omogočile, da izdelamo osebni model vsakega bolnika in prilagodimo specifična zdravljenja glede na njihove osebne značilnosti.

Do konca projekta namerava naš konzorcij:

- prepoznati genetske variante, povezane z občutljivostjo na AI, resnostjo in odzivom na zdravljenje
- oblikovanje modelov in vivo in in vitro za preiskave glavnih bioloških poti, na katere vpliva umetna inteligenca, in testiranje vpliva genetskih variant na imunsko in kožno celično biologijo
- izdelati zdravstveni karton za dopolnitev zdravstvene evidence z razvojem aplikacije za pametne telefone, ki bo na daljavo spremljala telesno in psihološko počutje bolnikov ter jim svetovala glede telesne aktivnosti ter prehranjevalnih in kadilskih navad
- predlaga nove metode stratifikacije, s katerimi lahko zdravniki ocenijo resnost, izberejo terapijo in sledijo izidu

Večina bolnikov čaka sedem let na diagnozo, kar predstavlja, če ni ustrezno zdravljeno, nevarnost za več sočasnih bolezni. V idealnem kliničnem obravnavanju je pri bolnikih treba hitro diagnosticirati in zgodaj razlojiti, da se lahko odločijo o svojem zdravljenju in da se jih lahko klinično spremlja za izboljšanje njihove kakovosti življenja.

Uporabniki bodo na dva načina lahko prispevali podatke:

- preko vprašalnikov, katere bodo izpolnjevali preko mobilne aplikacije ali preko spletne strani
- preko zbiranja podatkov njihove mobilne naprave (pametni telefon)

Zbiranje podatkov bo trajalo dlje časa. Za preverjanje, kako se situacija bolnika spreminja, se bodo vprašanja v vprašalnikih bolnikom ponavljala. Način vnosa bo narejen tako, da bo za uporabnika dovolj lahkoten, ter pogostost pojavljanja vprašanj bo dovolj redka.

1.2 Cilj projekta

Cilj projekta je, da partnerjem omogočimo poganjanje strojnega učenja na zbranih podatkih. Pri tem nam bo v pomoč program Orange, kateri omogoča na nek način poganjanje strojnega učenja preko uporabniškega grafičnega vmesnika, kot nasprotje računalniškega programiranja. V študiji bomo obravnavali tri vrste podatkov:

- Zdravstvene podatke bodo zbirali klinični partnerji (Köln, Milano, Innsbruck, Ljubljana), ali retroaktivno iz zdravstvenih kartonov ali prospektivno med pilotno študijo. Zdravstveni podatki bodo vsebovali klinične podatke (starost, spol, zgodovino bolezni in drugih komorbidnosti, zdravila, relevantne podatke za hidradenitis suppurativa – HS). Izbrane paciente bomo zaprosili za vzorce tkiva in slin (venska kri, lasni mešički).
- Genetske podatke bodo zbrali partnerji iz Trsta. Genetski podatki so sekvence nukleinskih aminokislin.
- Podatki o življenjskem slogu bodo zbrani direktno od pacientov z uporabo mobilne aplikacije, ki jo bomo razvili na Inštitutu Jožef Stefan. Zbirali bomo dva tipa podatkov: podatke, ki jih bodo uporabniki sami vnesli v aplikacijo (dolžina spanja, občutek bolečine, prehranske navade, kajenje, depresija, indeks dermatološke kakovosti življenja – DLQI) in podatke o fizični aktivnosti, ki jih bo z analizo

pospeškoma in drugih senzorjev v pametnem telefonu zbral računalniški algoritmi. Kjer bo mogoče, bomo uporabili standardne vprašalnike, ki jih priporočajo smernice (Dermatology life Quality index/DLQI), Hurley Score, Sartorius Score).

Poleg zgoraj omenjenih podatkov bomo s pomočjo algoritmov strojnega učenja poiskali nova znanja, gre za metapodatke in povezave med različnimi tipi podatkov.

1.3 Seznam partnerjev

- IRCCS materno infantile Burlo Garofolo, Italija
- Institute Klinik für Dermatologie und Venerologie, Nemčija
- Medical University Innsbruck, Avstrija
- Université Paris Est-Créteil, Francija
- Laboratory CNRS, Francija
- Institut "Jožef Stefan", Slovenija
- Fondazione IRCCS Ca' Granda-Ospedale Maggiore Policlinico, Italija

2 BACKEND - STREŽNIŠKI DEL

Arhitektura strežnika je sestavljena iz treh neodvisnih sistemov:

- strežnik, na katerem teče spletna stran
- strežnik za hrambo šifriranih podatkov
- strežnik za občasno varnostno kopiranje podatkov, ki sicer ni priključen v omrežje. Ta tip arhitekture nam omogoča višjo varnost.

2.1 Shranjevanje

2.1.1 Format in velikost baz. Podatke hranimo v različnih formatih, glede na vrsto podatkov.

- Zdravstveni podatki bodo shranjeni v excelovih tabelah ter bodo dodani v bazo tipa MongoDB prek spletnega vmesnika. Format bo usklajen s kliničnimi partnerji. Podatki bodo vsebovali številске in kategorične vrednosti.
- Genetski podatki bodo vsebovali številске in kategorične vrednosti. Podatki bodo obdelani z namensko programsko opremo, rezultati bodo shranjeni v formatih .xml ali .csv, kasneje v bazi MongoDB.
- Podatki o življenjskem slogu bodo raznoliki. Podatki, zbrani z vprašalniki, bodo številčni, kategorični ali opisnega tipa. Podatki o fizični aktivnosti bodo najprej obdelani na mobilni napravi, na strežnik se bodo pošiljali le agregirani podatki. Ti bodo v obliki v naprej izbranih opisnih značilk. Podatki se bodo zbrali v bazi MongoDB.
- Podatki, ki so lažje strukturirani, bodo shranjeni v bazi tipa MariaDB. To so npr.: podatki o uporabniških računih, geslih, pošti, nastavitvah, napravah ...

V okviru projekta bomo uporabili že obstoječe medicinske podatke pacientov, ki bodo sodelovali v študiji. Razlog je v tem, da se izognemo ponovnemu zbiranju dotičnih podatkov, kar bi bilo časovno zahtevno in drago. Podatke bodo priskrbeli klinični partnerji. Podatki bodo, tako kot novi podatki, anonimizirani.

2.2 Web API

API - application programming interface - je programski vmesnik, ki omogoča komunikacijo med dvema programi. Na našem strežniku teče API tipa REST. REST v večini primerov deluje zelo podobno kot spletne strani - teče na strežniku in čaka na klic odjemalca (client), kateremu vrača podatke iz podatkovnih baze.

Komunikacija poteka v večini primerov preko HTTP protokola in naša izvedba temu ni izjema. Strežnik je zunaj dosegljiv prek varnih API klicev.

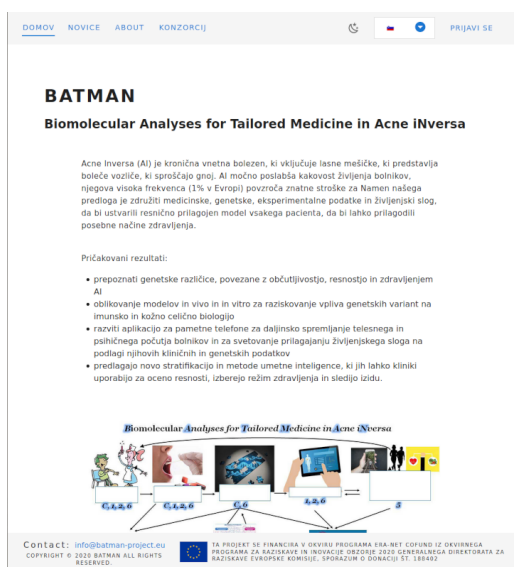
Strežnik deluje preko virtualnih Docker mikroservisov. Zgrajen je na osnovi PHP in Laravel.

API točke, ki jih izpostavlja naš strežnik (seznam mogoče še ni dokončen):

- Address API
- Authentication API
- Biometric API
- Device API
- Food databases list
- MongoDB
- News API
- Notifications API
- Questionnaire API
- Retrive user data API
- User Group API
- User Info API
- User Settings API
- User in Group API

3 FRONTEND - SPLETNA STRAN

Spletno stran smo izdelali z Vue.js, dodatkom za Vue.js Nuxt.js v načinu SSR, ter dodatkom za uporabniški vmesnik Vuetify.



Slika 2: Spletna stran kot je videti v tej fazi.

3.1 Metode strežbe spletnih strani

3.1.1 Static pages. Gre za preprosto stran brez dinamične vsebine. Strani so narejene v obliki html datoteke in so postavljene na strežnik. Statične strani ni treba nič spreminjati, ker ostane vedno ista. Strežnik s tako stranjo nima posebnega dela.

Take strani niso primerne za strani z vsebino, ki jo spreminja uporabnik, npr. spletni forumi, spletne trgovine, strani, kjer uporabnik izbira, kakšno vsebino bi rad videl (tipično podatki iz podatkovne baze). Slabost te metode je, da se za vsako stran mora prenesti ustrezno html datoteko iz strežnika in jo potem mora spletni brskalnik (v nadaljevanju: brskalnik) odjemalca izrisati na

stran (document object model), kar negativno vpliva na gladkost delovanja strani.



Slika 3: Odjemalec prejme od statičnih in dinamičnih strani več kot eno html datoteko

3.1.2 Dynamic pages. Idealno za npr. spletno trgovino. Uporabnik s svojimi dejanji (npr. polnjenje košarice v spletni trgovini) spreminja vsebino strani. Ta vsebina se predpripravi na strežniku, kateri, preko s programskim jezikom (običajno Python, PHP ali Java), pripravi prilagojeno html datoteko s končno vsebino. Ta datoteka se potem prenese do odjemalca. Spletni pajki (web crawlers), ki jih spletni iskalniki poganjajo, da raziskujejo splet brez težav preberejo vsebino takih strani, zaradi česar je naša stran bolj zgodaj prikazana med zadetki iskanja, kar je to kar želimo. Slabost te metode je podobna kot pri statičnih straneh; celotno stran je treba pogosto znova nalagati naložiti in uporabniška izkušnja ne uspe dati občutka, da deluje gladko kot mobilna aplikacija.

3.1.3 Single Page Application. Single Page Application je aplikacija, ki deluje v brskalniku in med uporabo ne zahteva ponovnega nalaganja strani. To vrsto aplikacij uporabljate vsak dan. To so na primer Gmail, Google Maps, Facebook ali GitHub. Prednosti SPA so, da proizvedejo izjemno uporabniško izkušnjo tako, da poskušajo posnemati "naravno" okolje v brskalniku - brez nalaganja strani, brez dodatnega čakalnega časa. Samo ena spletna stran, ki jo obiščete, nato naloži vso drugo vsebino z uporabo JavaScripta. Popularni JavaScript okvirji za SPA so Angular, React, Vue.js,



Slika 4: SPA: odjemalec prejme eno html in JavaScript datoteko

Ember.js, Meteor.js, Knockout.js.. Spletna mesta z eno stranjo pomagajo uporabniku, da ostane v enem, udobnem spletnem prostoru, kjer je vsebina uporabniku predstavljena na preprost, enostaven in izvedljiv način[3]. Html datoteko (Document Object Model) ustvari brskalnik. SPA je sodoben način izdelave spletne strani, in tak tip strani se izdelava z uporabo vseh štirih najpopularnjših sodobnih ogrodij, to so Angular, React, Vue.js, Ember. Prednosti:

- poti znotraj strani (routes)
- obnaša se podobno kot program, brez ponovnih nalaganj (reloading)
- brskalnik jih zlahka shrani v svoj predpomnilnik (cache)
- delujejo hitro
- spremeni se samo del strani, kjer je prišlo do sprememb
- vsebina se naloži samo enkrat

Slabosti:

- čas nalaganja prve strani je daljši
- običajno deluje počasneje na počasnih povezavah ali počasnih napravah

- vsebina strani je težje dostopna črvom od spletnih iskalnikov

Skratka, z SPA aplikacijo so vse zahteve preusmerjene v korenko datoteko - prva html datoteka, nato pa ogrodje preusmeri uporabnika z vgrajenim sistemom usmerjanja. Nalaganje prve strani je torej vedno ista korenka datoteka.[5]

3.2 Server Side Rendering

Velja za metodo ki proizvede najbolj odzivno spletno stran. Strežnik vrne statično html datoteko, iz katere potem gre naložiti ostale vsebine kot so CSS datoteke, fotografije, JavaScript datoteke ... Kadar brskalnik naloži CSS datoteke, lahko izriše stran, tudi če še niso naložene fotografije in JavaScript datoteke. Ta proces je običajno zelo hiter. Zadnji del procesa nalaganja je JavaScript. Ta lahko traja relativno dolgo, če je datotek več ali če so velike. Brskalnik mora potem te datoteke še interpretirati. Če ima uporabniška naprava nizko zmogljivost strojne opreme, je ta proces dolgotrajen. Če je prvi izris strani odvisen od JavaScripta, je čas nalaganja strani posebej počasen. Če želimo imeti hitro nalaganje prve strani, mora naša spletna stran imeti dovolj izrazito html in css datoteko. Na JavaScript bi morali gledat kot izboljšanje html-ja in CSS-ja, ker ga obravnavamo pozneje. V sodobnem svetu frontenda, kjer se uporabljajo ogrodja kot so Angular, React, Vue.js, to žal ni več enostavno, ker take strani nujno potrebujejo JavaScript da sploh prikažejo kaj vsebine. Html datoteka (torej to kar potem postane štran) se izdela na strežniku. Ko pajki iskalnikov pregledajo stran, je stran že zapolnjena z vsebino in meta podatki. Omogoča tudi hitrejši prikaz strani, ker že prihaja s strežnika v večinomoma popolni obliki. Najpopularnejša orodja za SSR so Next.js (v njegovi različici Nuxt.js) in Gatsby.

3.2.1 Vue.js. Vue.js, progresivni JavaScript okvir, ki nam omogoča hiter razvoj enodelnih aplikacij (SPA), ki jih je enostavno vzdrževati ob uporabi njegovih komponent in reaktivnih funkcij, kot so reaktivni vmesniki in vezava podatkov. Ustvaril Evan You, Vue.js je odprtokodni progresivni JavaScript okvir za gradnjo uporabniških vmesnikov (uporabniških vmesnikov) in aplikacij za eno stran; običajno ga imenujemo kar Vue. Ta okvir uporablja "visoko ločitev", ki razvijalcem omogoča postopno ustvarjanje uporabniških vmesnikov (UI).[4]

V primerjavi z Angular in React, je Vue.js najbolj primeren za manjše "projekte". Velikost sproducirane spletne strani v bajtih je z Vue.js najmanjša izmed te trojice. Vue.js je tudi najenostavnejši za uporabo, kar pohitri razvoj, še posebej če programirajo ne samo senior ampak tudi mid in junior programerji.

3.2.2 Nuxt.js. Nuxt.js je ogrodje za ustvarjanje aplikacij Vue.js. Njegov cilj je pomagati razvijalcem Vue, da hitro, enostavno in organizirano izkoristijo vrhunske tehnologije.

Nuxt.js je brezplačni in odprtokodni okvir spletnih aplikacij, ki temelji na Vue.js, Node.js, Webpack in Babel.js (zgleduje se po Next.js [4]). Okvir se oglašuje kot "metaokvir za univerzalne aplikacije". Tu se uporablja izraz univerzalni, kar pomeni, da je cilj ogrodja omogočiti uporabnikom, da ustvarijo spletne poglede v JavaScriptu z uporabo znanega sistema enojnih komponent Vue.js [5] in ki lahko deluje kot enostranska aplikacija v brskalniku. (SPA) pogledi, kot tudi spletni pogledi, upodobljeni s strežnikom, ki se nato (po upodabljanju strežnika) "rehidrirajo" v popolno funkcionalnost SPA [6] [7]. Poleg tega ogrodje uporabnikom omogoča, da imajo vsebino ali njene dele v celoti predhodno

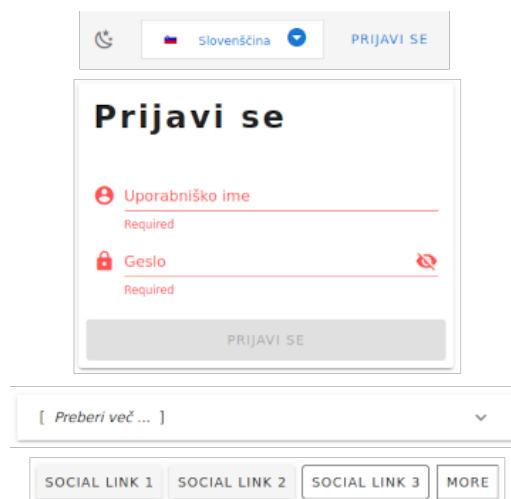
upodobljene na strežniku in strežene na način statičnih generatorjev spletnih mest.

Prednosti tega pristopa so med drugim tudi skrajšanje časa za interaktivnost in izboljšanje SEO v primerjavi s SPA-ji, ker spletni strežnik streže celotno vsebino vsake strani, preden se izvede kateri koli JavaScript na strani odjemalca. Druga možnost je, da lahko ohranite prednosti tradicionalnih HTML strani na strežniku in izboljšate interaktivnost in napredni uporabniški vmesnik POV. Glavna prednost samega okvira Nuxt.js je ta, da konfiguracijo in nastavitve takih aplikacij poenostavi in omogoči brezhibno razvijanje programov, ki lahko preprosto razvijejo dele uporabniškega vmesnika aplikacije, kot da bi šlo za bolj pogosto datoteko Vue.js aplikacijo. [2]

Kombinacijo SSR in SPA imenujemo universal application in v tem načinu smo naredili spletno stran. Spletno stran smo naredili v petih jezikih - to so jeziki partnerjev projekta. Temu se v programiranju reče i18n, kar je nekako okrajšava za internationalization (i, 18 črk in n). Za to smo uporabili knjižnico Nuxt-i18n.

Spletna stran omogoča novim uporabnikom registracijo ter obstoječim uporabnikom prijavo na spletno stran - na svoj profil. Za shranjevanje uporabniških preferenc, poleg podatkov na strežniku, smo uporabili knjižnico vuex-persist.

3.2.3 Vuetify. Vuetify je knjižnica za uporabniški vmesnik. Osnovana je na Googlovem "material design" dizajnu, ki je nastal leta 2014 in je danes v polnem zagonu. Google uporablja material design na svojih aplikacijah in spletnih straneh. Vuetify ponuja že narejene komponente, katere za izgled uporabljajo material design. Poleg tega ponuja tudi mrežni sistem za postavitve elementov na spletni strani (grid system). Ta edinstven pristop omogoča uporabnikom izredno intuitivno uporabo storitev. Cilj material design projekta je uporabnikom zagotoviti vse, kar je potrebno za izdelavo bogatih in privlačnih spletnih aplikacij s specifikacijo Material Design.



Slika 5: Primeri material design iz spletne strani projekta

V projektu smo uporabili Vuetifyjeve module kjer se je dalo, to nam je omogočilo skrajšat čas razvoja. Pomembno je tudi to, da tej moduli imajo skladen design, zaradi česar design na strani deluje bolj poenoten.

V skladu s sodobnimi smernicami, je spletna stran narejena z dvema izgledi - svetel in temen način. Za to smo kar uporabili

Vuetifyjevo rešitev za upravljanje z izgledi.

Kot je danes standard, uporabljamo na spletni strani teste. Za to smo uporabili ogrodje Jest.

4 STROJNO UČENJE IN ORANGE

Orange je orodje (program) za delo z "big data". Podpira veliko interaktivnih vizualizacij. Njegov grafično uporabniški vmesnik omogoča "vizualno" programiranje. S tem se lahko ukvarjajo z strojnimi učenjem tudi ljudje, ki niso večji računalniškega programiranja. Orange ni spletna storitev. Vsakdo ki ga želi poganjati si ga mora namestiti na svoj računalnik. Ideja je, da bodo partnerji lahko poganjali Orange na podatkih, zbranih v tej raziskavi. Za pomoč jim bomo na IJS izdelali predloge.

5 ZAKLJUČEK

Projekt je v drugem letu življenja od predvidenih treh. Čeprav informacijsko-tehnološki del projekta še ni zaključen, smo lahko že predstavili njegovo izvedbo. Podali smo strokovno razlago za naše odločitve na področju frontenda (spletna stran). V roku enega meseca se bodo predvidoma začeli nabirati resnični podatki pacientov. Ko bo podatkov dovolj, se bo lahko poganjalo strojno učenje nad podatki, kar je tudi glavni doprinos projekta družbi. Skratka, izboljšalo se bo raziskave o Acne Inversa.

ZAHVALA

Ta projekt se financira v okviru programa ERA-NET Cofund iz okvirnega programa za raziskave in inovacije Obzorje 2020 Generalnega direktorata za raziskave Evropske komisije, Sporazum o donaciji št. 188402.

Zahvaljujemo se tudi študentom, ki so sodelovali na projektu, to so Laura Guzelj Blatnik, Tine Rozmanič, Lučka Šnofl in Žan Špiler.

LITERATURA

- [1] Br J Dermatol Desai N Shah P. 2017.
- [2] <https://en.wikipedia.org/wiki/Nuxt.js>. 2020.
- [3] <https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58>. 2016.
- [4] <https://www.avantica.com/blog/what-is-vue.js-and-how-do-we-use-it>. 2019.
- [5] <https://www.toptal.com/front-end/client-side-vs-server-side-pre-rendering>. 2020.
- [6] J Am Acad Dermatol Jemec GB et al. 1996.
- [7] J Cutan Med Surg Mehdizadeh A et al. 2018.