**IDA-VIRUMAA KUTSEHARIDUSKESKUS**

**Eriala: noorem tarkvaraarendaja**

**Manga Man Projekt**

**Timo Häkkinen ja Mihhail Levin**

**Juhendaja: Aleksander Pulver**

**Jõhvi, 2018**

**SISSEJUHATUS**

Eestis pole ühtegi kauplust, kus oleks müüdud Jaapani manga-koomikseid, nii et meie ideeks oli teha pood, kus kasutajal on palju võimalusi. Meie poes on võimalik füüsiliselt saada raamatu koopiat koos e-raamatuga (PDF), ning on veel võimalus tasuta saatmine ülemaailmselt. Raamatuid on ka erinevates keeltes, näiteks: inglise, vene, itaalia, jaapani ja paljudes muudes keeltes. Klienditugi on meil samuti olemas juhuks, kui meie klientidel tekivad küsimused. Klientidel on võimalus end registreerida ja muuta oma andmeid.

Määratud tööülesanded meie projektis on järgmised:

* Võimalus lisada erinevaid raamatuid ja raamatuosi kiiresti ja efektiivselt.
* Võimalus muuta raamatute andmeid.
* Võimalus lisada raamatuid ostukorvi.
* Võimalus teha raamatutele allahindlusi.
* Kasutaja registreerimise võimalus.
* Kasutaja sisselogimise võimalus.
* Kasutaja enda andmete muutmine.
* Võimalus Kasutajal saata kiri administraatoritele.
* Administraatori sisselogimise võimalus.
* Administraatoril raamatute lisamise võimalus.
* Administraatoril võimalus teha muudatusi andmebaasis administraatori paneeli kaudu.

Antud töö eesmärgiks on luua raamatupood valides paljude võimaluste seast välja lihtsamad ja mugavamad meetodid. Projektis kasutame programmi XAMPP koos Apache ja MySQL moodulitega. Päringute ja andmebaasi sidumine on kirjutatud põhiliselt PHP-ga ilma mingi raamastikuta. Veebilehe kujundus ja väljanägemine oli kasutatud raamastik BootStrap 4.0, lisaks ja JavaScript-ile kasutatud raamastikud on jQuery ja Popper.js. Andmete edastamiseks kasutasime GitHub-i ja tegime avaliku hoidla, kus me saame muuta ja lisada faile.

Timo Häkkinen on projektijuht ja front-end developer, tegeleb projekti juhtimisega ja kujundusega.

Mihhail Levin on back-end developer, tegeleb andmebaasi loomisega ja funktsionaalsusega.

**MÕISTED**

PHP (*Hypertext Preprocessor*) - skriptimiskeel, mida kasutatakse peamiselt serveripoolsetes lahendustes dünaamiliste veebilehtede loomisel. PHP toetab objektorienteeritud programmeerimist, andmebaasidega suhtlemist ning moodulite kaudu veel paljusid protokolle ja rakendusi[1].

HTML (*Hyper Text Markup Language*) – veebilehtede märgendamiskeel.

CSS (*Cascading Style Sheets*) – keel, mida kasutatakse xHTML ja HTML lehtede kujundamiseks.

BootStrap – on HTML-i ja CSS-i raamastik, mis teeb veebilehe kujundamise kergemaks ja kiiremaks.

jQuery – on *JavaScript*’i raamatukogu, mis on välja töötatud kliendipoolse ja kujunduse skriptimise hõlbustamiseks.

Popper.js - on *JavaScript*’i raamatukogu, kujunduse skriptimise hõlbustamiseks.

JavaScript – on objektorienteeritud programmeerimiskeel, mida kasutatakse peamiselt HTML lehekülgede skriptimisel.

MySQL – relatsioonilise andmebaasi haldamise süsteem, millesse saab päringud esitada standardiseeritult[2].

URL (Uniform Resource Locator) – Internetiaadress ehk universaalne ressursilokaator ([ingl](https://et.wikipedia.org/wiki/Inglise_keel" \o "Inglise keel) uniform resource locator, URL) on [infotehnoloogias](https://et.wikipedia.org/wiki/Infotehnoloogia" \o "Infotehnoloogia) ühene aadress, mida kasutatakse infoallikate leidmiseks ja kasutamiseks [internetis](https://et.wikipedia.org/wiki/Internet)[3].

XAMPP ( Cross-platform, Apache, MariaDB, PHP, Perl) – On lihtne ja kerge Apache jaotus, mis muudab arendajatele väga lihtsaks katsetamiseks ja kasutuselevõtuks kohaliku veebiserveri loomise.

**ANALÜÜS**

Analüüsi osas toome välja võrdlused ja põhjused, miks just nende kindlaid meetodeid kasutame. Alustades raamastiku valikuga

***Raamastiku valik***

Raamastiku valikul vaatasime raamastiku lihtsust, mugavust, ning põhjaliku dokumentatsiooni olemasolu. Samuti võtsime arvesse populaarsust ja suurust ning teisi omadusi, mis on toodud tabelis[4].

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Omadus | Bootstrap | Foundation | Bulma | Ulkit | Semantic UI |
| Praegune versioon | 4.0.0 | 6.4.3 | 0.6.0 | 3.0.0 | 2.2 |
| Populaarsus | 116 658 tähte GitHub-is | 26 455 tähte GitHub-is | 20 325 tähte GitHub-is | 10 761 tähte GitHubis | 37 594 tähte GitHub-is |
| Põhikontseptsioonid/ põhimõtted | RWD ja mobile first | RWD, mobile first ja semantic | RWD, mobile first ja modern | RWD ja mobile first | Semantic, tag ambivalence ja responsive |
| Eelprotsessorid | Less ja Sass | Sass | Sass | Less Ja Sass | Less |
| Suurus | 154 KB | 195.5 KB | 184 KB | 326.9 KB | 806 KB |
| Dokumentatsioon | 4/5 Hea | 4/5 Hea | 4/5 Hea | 4/5 Hea | 4.5/5 Väga Hea |
| Regeeriv | Jah | Jah | Jah | Jah | Jah |
| Modulaarne | Jah | Jah | Jah | Jah | Jah |
| Starteri-mallid | Jah | Jah | Jah | Jah | Jah, lihtsamad starteri-mallid |

Seletused

* RWD (Responsive Web Design) – on, et veebileht näeks välja hästi kõikidel seadmetel.
* Mobile first – Mobiiltelefonidele reageeriv.
* Semantic – Toetab „dokumentidele veebis“ andmebaasides leiduvaid andmeid.
* Starteri-mallid – mallid, mis arendaja saab kasutada, et läbida kiiremini oma projekti.
* GitHub – on arendusplatvorm, mis on inspireeritud teie töötavast. Avatud lähtekoodist ettevõttele saate paigutada ja vaadata koodi, hallata projekte ja ehitada tarkvara koos miljonite teiste arendajatega.
* Less (Leaner Style Sheets) – CSS-i jaoks on tagurpidi ühilduv keele laiendus.
* Sass (Syntactically Awesome Style Sheets) – mida, kasutatakse [veebilehtede](https://et.wikipedia.org/wiki/Veebileht) kujundamiseks, sassis kirjutatud kood teisendatakse edasi [CSS-i](https://et.wikipedia.org/wiki/CSS).

***Andmebaas***

Andmebaasi meil polnud suur valikut, sellepärast meie kasutasime XAMPP-i MySQL-i, mis oli sisseehitatud XAMPP-i koos phpmyadmin-iga. Seni kuni MySQL on ka kõige populaarsem ja lihtsam andmebaas, see on ka üks põhjustest, miks meie valisime MySQL.

***Vahendite valik***

Koodi kirjutamiseks me kasutasime programmi Atom, koos mõne pluginaga, et teha koodimine kergemaks. Veebilehe disaini tegemiseks kasutasime Adobe Photoshop CC 2015. Logo tegemiseks kasutasime https://vectr.com/ online programmi. Failide edastamiseks kasutasime programmi GitHub Desktop, kus meie esitame muudatusi oma projektile ja vaatame üle muudatusi, kes mida tegi. Omavaheliseks suhtlemiseks kasutasime programme Discord ja Steam.

***Rakenduse struktuur***

Rakenduse struktuur on küllaltki tavaline, üleval asub navigatsiooni paneel, kus kasutaja saab valida millisele leheküljele ta tahab astuda. Tavakasutaja, registreeritud kasutaja ja administraator näevad navigatsioon paneeli erinevalt, aga ma selgitan struktuuri administraatori vaatenurgast. Esilehel on meil tekst ja pilt, et kasutaja saaks tutvuda meie veebilehega. Nende all on parimad müüjad ja uued raamatud. Shop-i leheküljel on kõik meie raamatud ja on antud kasutajale oma otsingumootor, kus ta saab valida hinda, keelt, žanri ja muud. Sale-i lehel on kõik raamatud, mis on meil hetkel allahindlusega. About-i lehel on informatsioon ettevõte ja meie kohta. Support-i lehel on vormid registreeritud kasutajatele küsimuse või kaebuse esitamiseks. Dashboard on administraatori paneel, kus ta saab lisada, muuta ja kustutada raamatuid, ja on võimalus vaadata üle kasutajatelt saabunud kaebusi ja küsimusi. Kõige paremal on meil kiirotsingumootor, ostukorv ja kasutajate leht, kus nad saavad muuta oma andmeid, näiteks kasutajanime, parool ja e-mail.

***Kujundus***

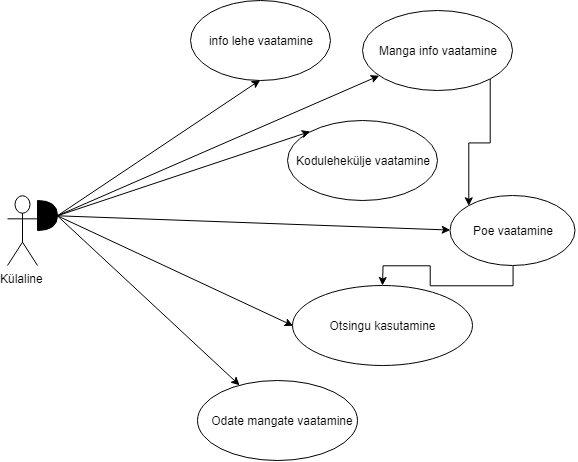
Kujunduse pool on tehtud kõik BootStrap-iga. BootStrap kasutab grid süsteemi, mis annab meile võimaluse teha veebilehte kiiresti ja efektiivselt. Raamatud on tehtud card-i süsteemiga, mis teeb kompaktseid kaardikujulisi juppe, kuhu saab panna pildi, sisu ja footer, ehk siis raamatu pilt, nimi, autor ja hind. Administraatori paneelis on tehtud andmete vaade tabelis ja lisamine, muutmine ja kustutamine on tehtud modal-iga, modal on käsk, mille nupu vajutamisel, tuleb välja vorm, kus administraator saab muuta raamatu andmeid.

**Struktuur**

Järgnevalt esitletakse kasutusjuhtude diagrammid, mis on koostatud põhifunktsioonide kaupa ning tuuakse välja andmebaasi ER(Entity-relationship) skeem ning objektide ja atribuutide semantika.

Külalise vaade

Külalise vaate kasutusjuhtude diagramm.



Kasutaja vaade

Kasutaja kasutusjuhtude diagramm.



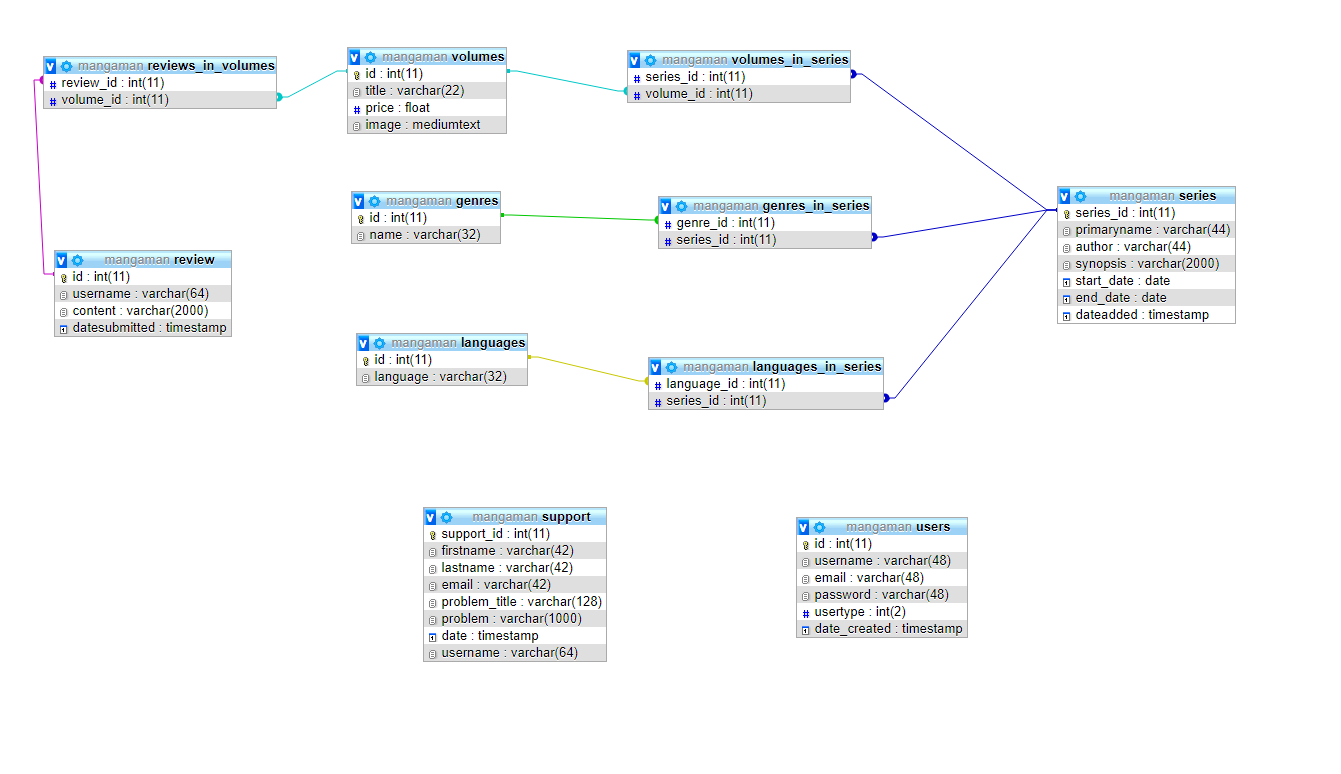
Administraatori vaade

Administraatori kasutusjuhtude diagramm.



ER skeem

Andmebaasi ER skeem.



Objektide ja atribuutide semantika

Tähistused

PK – *Primary Key*

FK – *Foreign Key*

Objektide ja atribuutide semantika

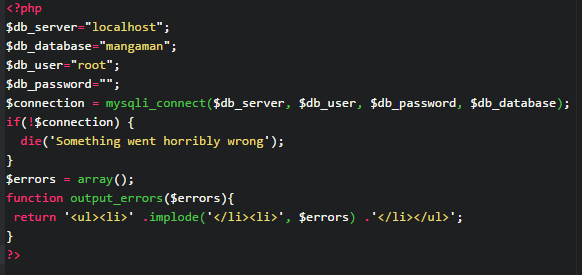
|  |  |
| --- | --- |
| Nimetus | Kirjeldus |
| series | Sarja pealkirja ning detaile kuvatakse kodulehel ja manga detailide lehel |
| series\_id(PK) | Unikaalne identifikaator |
| primaryname | Manga sarja pealkiri |
| author | Manga sarja autor |
| synopsis | Sarja |
| start\_date | Manga sarja alustamis kuupäev |
| end\_date | Manga sarja lõpetamis kuupäev |
| dateadded | Sarja lisamise aeg |
| volumes\_in\_series | Ühendamis tabel, mis ühendab series ja volumes kokku |
| series\_id(FK) | unikaalse series identifikaatori ühendus volumes unikaalse identifikaatoriga |
| volume\_id(FK) | unikaalse volumes identifikaatori ühendus unikaalse series identifikaatoriga |
| volumes |  |
| id(PK) | Unikaalne identifikaator |
| title | Sarja osa identifikaator |
| price | Sarja osa hind |
| image | Manga sarja osa pilt |
| reviews\_in\_volumes | Ühendamis tabel mis ühendab kommentaarid osadega |
| review\_id(FK) | unikaalse review identifikaatori ühendus volumes unikaalse identifikaatoriga |
| volume\_id(FK) | unikaalse volumes identifikaatori ühendus unikaalse review identifikaatoriga |
| review | Seda kuvatakse manga osade all |
| id(PK) | Unikaalne identifikaator |
| username | Kasutaja nimi kes seda |
| content | Kommentaari sisu |
| datesubmitted | Kommentaari lisamise aeg ja kuupäev |
| genres\_in\_series | Ühendamis tabel, mis ühendab žanrid sarjadega |
| genre\_id(FK) | Unikaalse genre identifikaatori ühendus series identifkiaatoriga |
| series\_id(FK) | Unikaalse series identifikaatori ühendus genre identifkiaatoriga |
| genres | Žanride tabel mille kuvatakse manga detaili lehel |
| id(PK) | Unikaalne identifikaator |
| name | Žanri nimi |
| languages\_in\_series | Ühendamis tabel, mis ühendab keeled sarjadega |
| Language\_id(FK) | Unikaalse language identifikaatori ühendus series identifkiaatoriga |
| Series\_id(FK) | Unikaalse series identifikaatori ühendus language identifkiaatoriga |
| Languages | Keelte tabel mida kuvatakse manga detaili lehel |
| Id(PK) | Unikaalne identifikaator |
| Language | Keele nimi |
| Users | Kasutajate tabel |
| Id(PK) | Unikaalne identifikaator |
| Username | Kasutaja nimi |
| Email | Kasutaja lisatud e-maili aadress |
| password | Kasutaja parool |
| userype | Kasutaja tunnus, ehk kas on admin või mitte |
| Date\_created | Kasutaja registreerimise aeg |
| Support | Kontakti vormi tabel |
| Support\_id(PK) | Unikaalne identifikaator |
| firstname | Kasutaja eesnimi |
| lastname | Kasutaja perenimi |
| Email | Kasutaja email |
| Problem\_title | Kontakti teema |
| Problem | Sõnum |
| date | Andmebaasi lisamise aeg |
| username | Kasutaja nimi |

**Realiseerimine**

Lehekülje kirjutamiseks pidime seadma testimiskeskkonna, milleks sai xampp ja google chrome. Alustasime esialgse kujunduse paika panemisega ning siis alustasime php kirjutamisega.

Andmebaasi konfiguratsioon

Meie andmebaasi konfiguratsioon asub failis nimega config.php, et oleks ühendus valitud andmebaasiga. Failis peab olema hosti nimi, kasutaja nimi, andmebaasi nimi, kasutatav parool jne. Kogu fail on päris väike siin on näide:

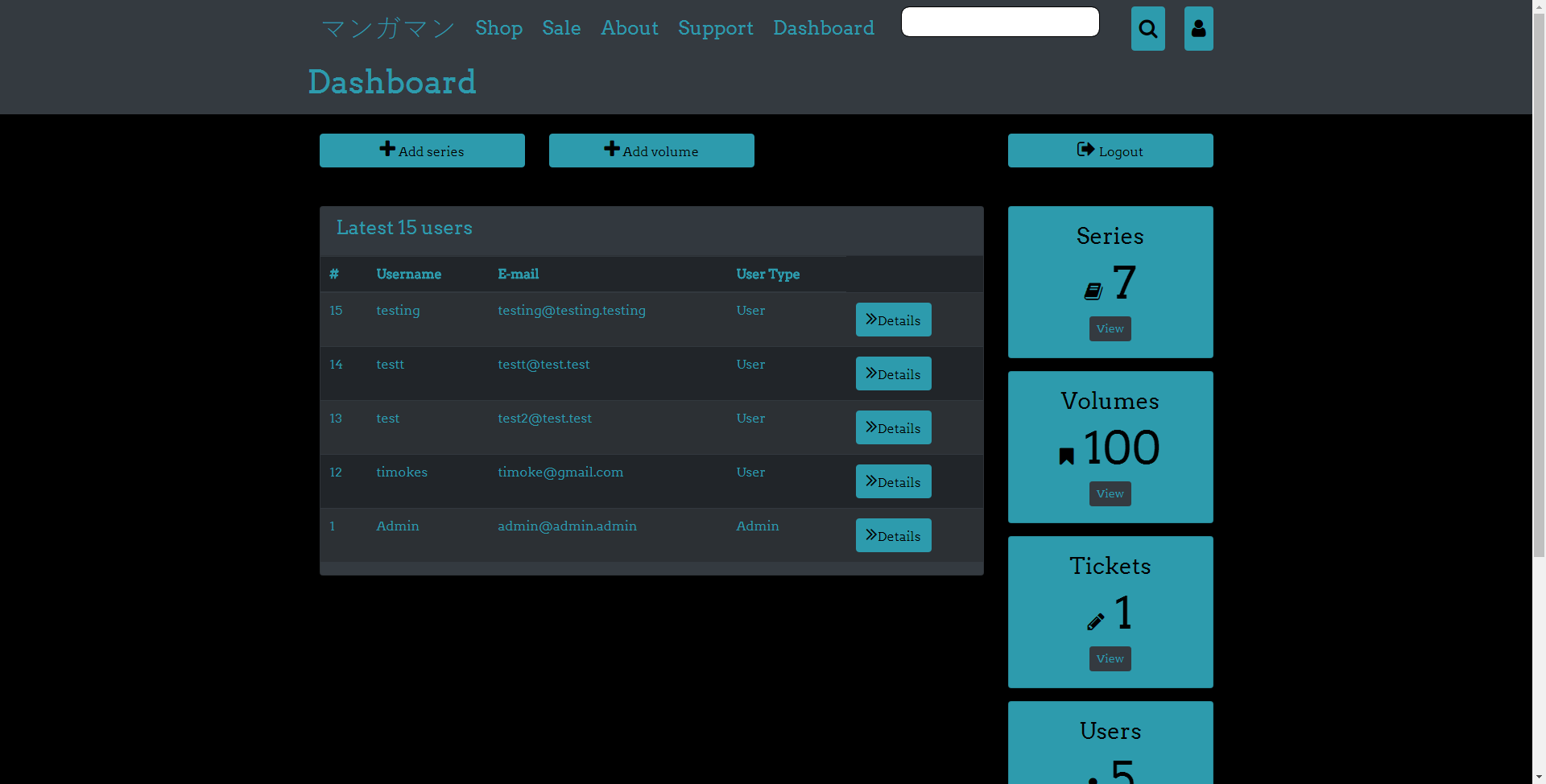


Administraator

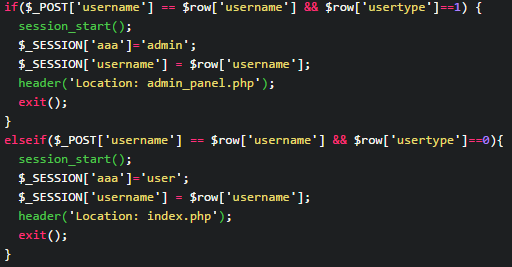
Administraatori lehekülje kättesaamiseks on vajalik logida sisse admini kasutajaga. Seda kasutaja tunnust on võimalik määrata ainult manuaalselt andmebaasis endas. Administraatori autentiseerimisel on temal isiklik navigatsiooni header milles on küljes uus nupp mis viib administraatori paneelile.

Administraatori autentiseerimine toimub sessioonidega ning andmebaasi ‘users’ tabeli ‘usertype’ rea kontrolliga, juhul kui see vastab 1-le saadab logimise leht kohe adminipaneeli, kui see vastab 0-le sis on tegemist tava kasutajaga, keda saadetakse kodulehele.

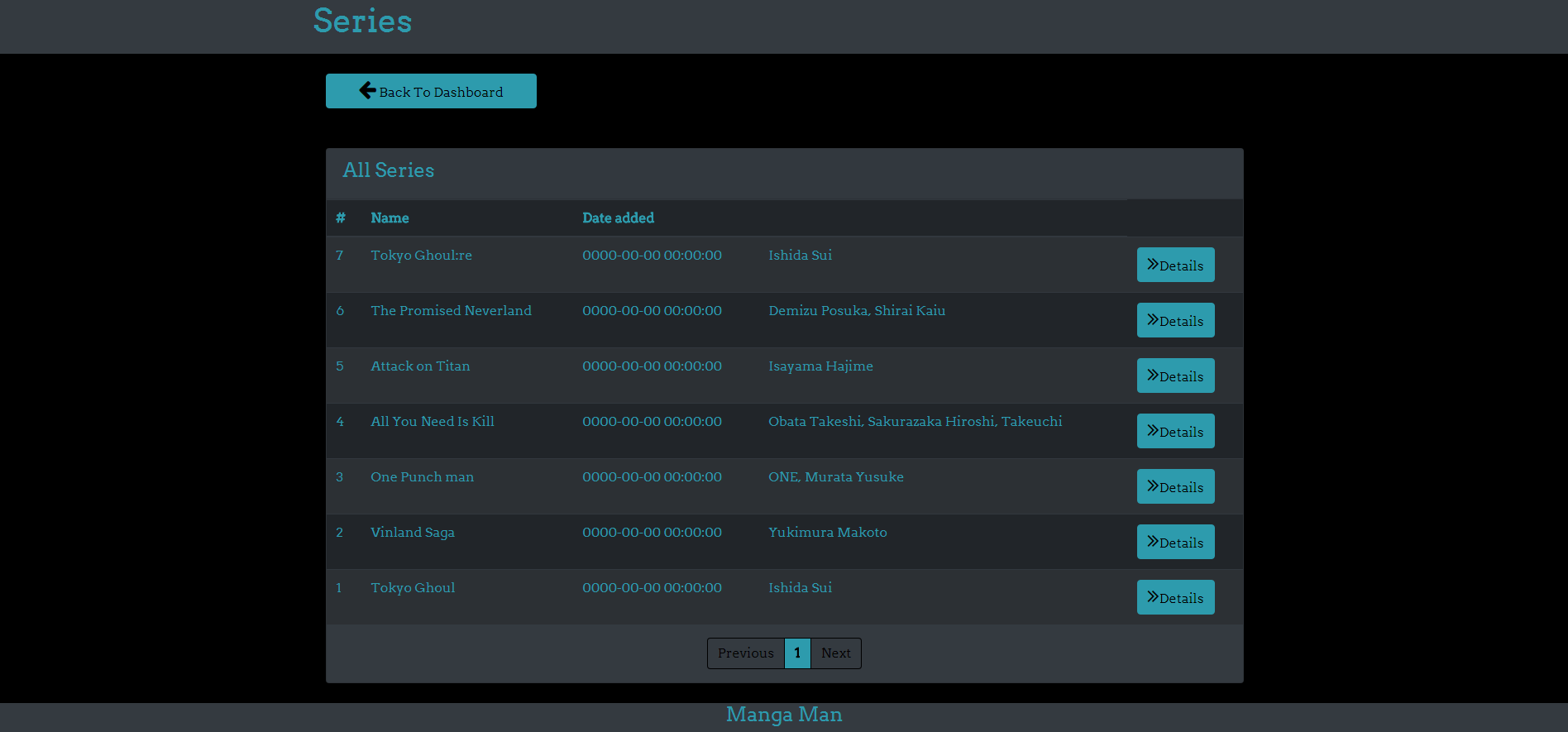
Admini paneeli vaade:



Siin on väike näite sellest kontrollist:

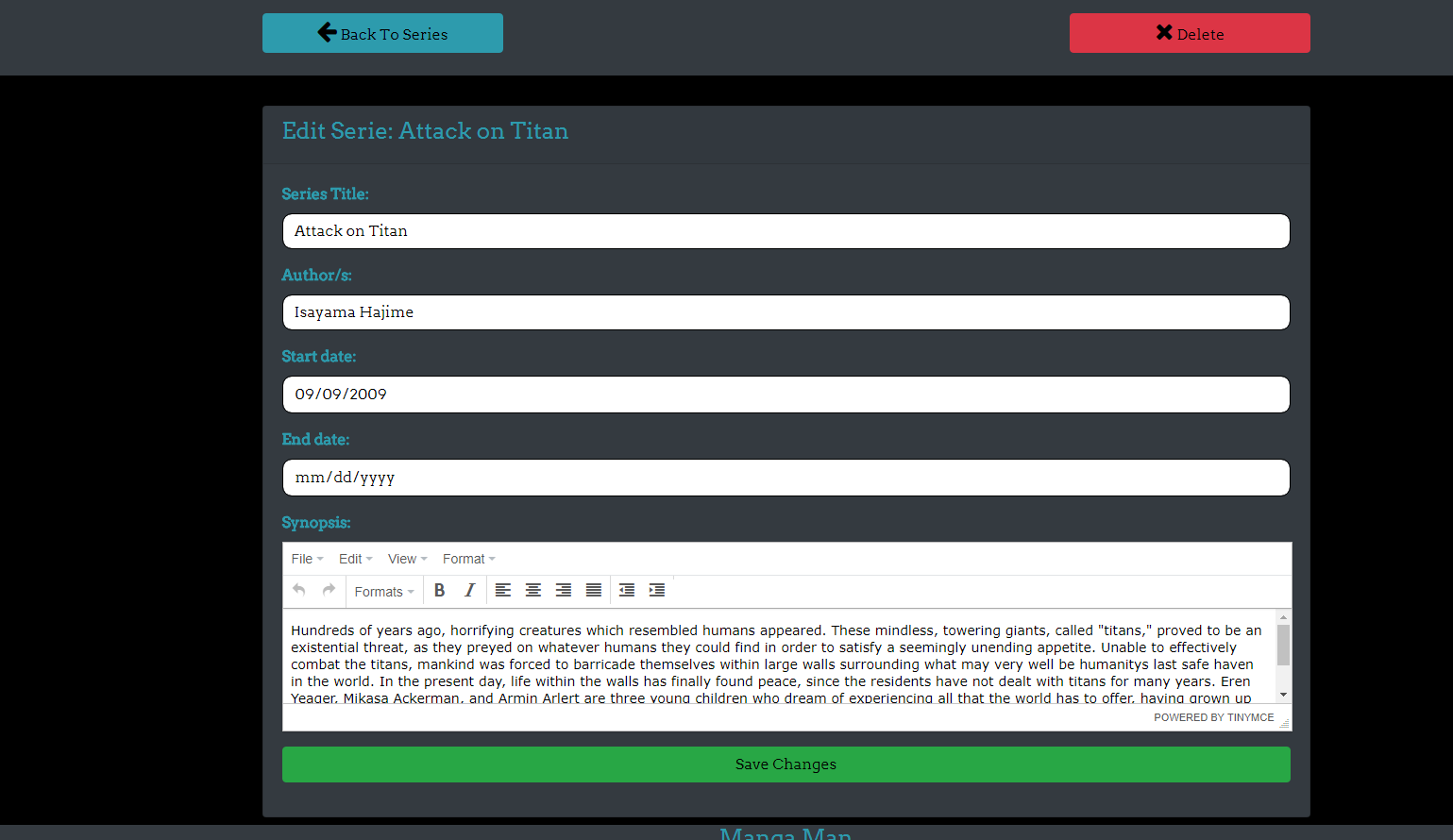


Lähemalt administraatori paneeli struktuuriga, kasutame CRUD (Create Update Delete) vaateid. Administraatori paneelil on mitu linke nagu näha pildil, mis viivad vaatesse kus saab teha erinevaid muudatusi andmebaasis. Näiteks sarjade lisamiseks/muutmiseks/kustutamiseks läheks administraator ’series’ alt ’view’ klipsades. Sealt näeks ta midagi sellist:

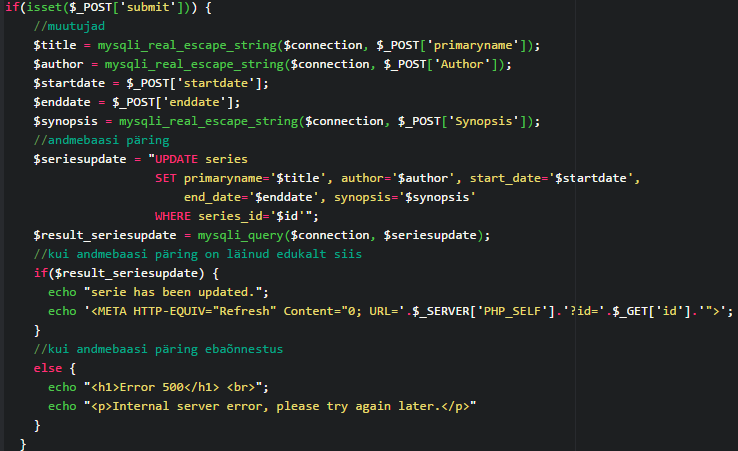


Klipsades ’details’ nuppu saab administraator selle manga sarja muuta või kustutada.

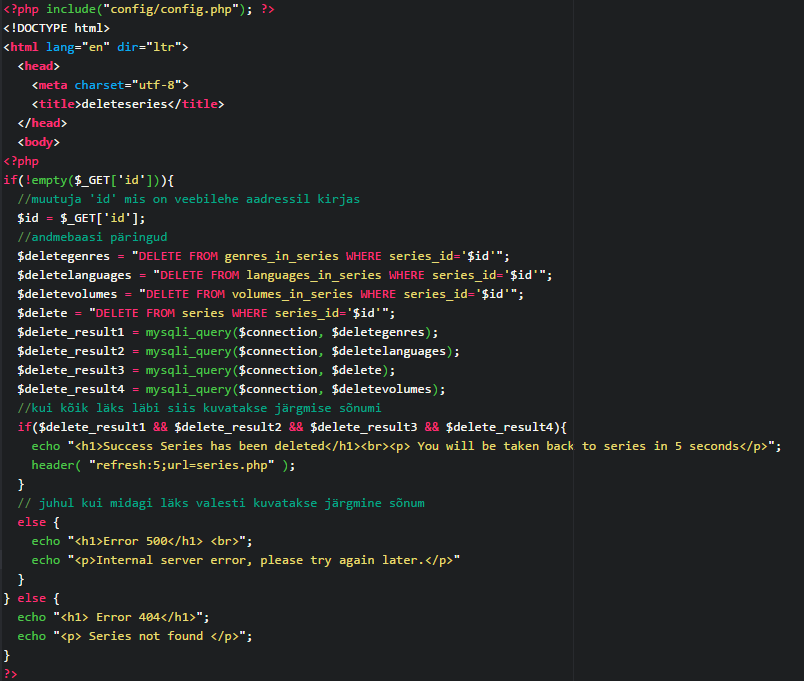
Näite detailidest:



Koodi näide detailide muutmisest:



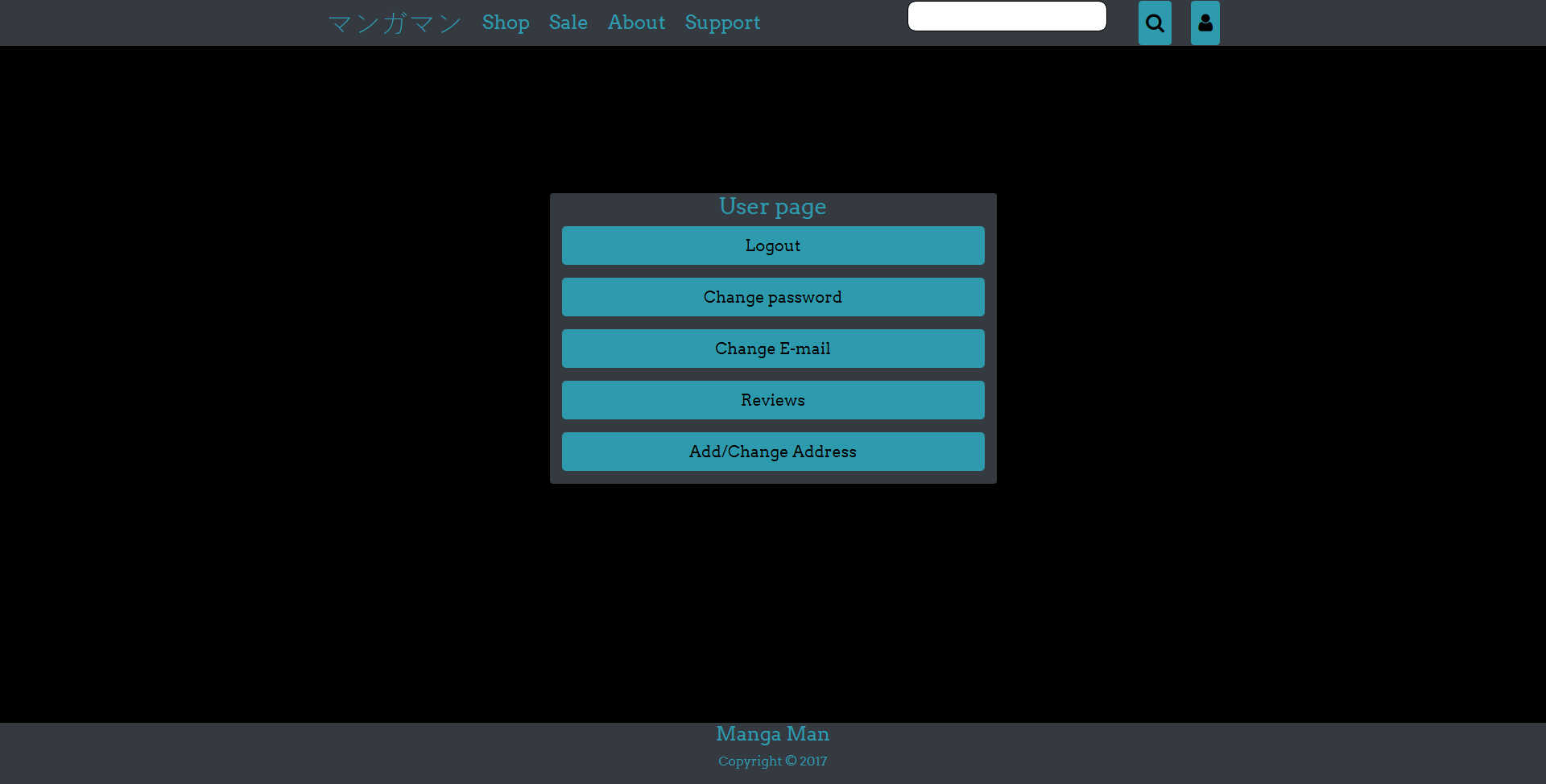
Koodi näide sari kustutamiseks:



Kasutaja

Kasutaja logimisel toimub samasugune kontroll nagu administraatoril, eelmisel näidel on näha, kui andmebaasi ’users’ tabeli ’userype’ rida vastab 0 siis sessiooni ’aaa’ seadistatakse ’user’-ks ehk tava kasutajaks. Tavakasutajal on ka enda navigatsiooni header, milles on jällegi rohkem nuppe. Samas on kasutajal ka enda väike lehekülg kus ta saab muuta enda parooli, emaili ning aadressi. Samamoodi võib kasutaja ka muuta enda kommentaarid mida tal on võimalus lisada iga manga osa kohta.

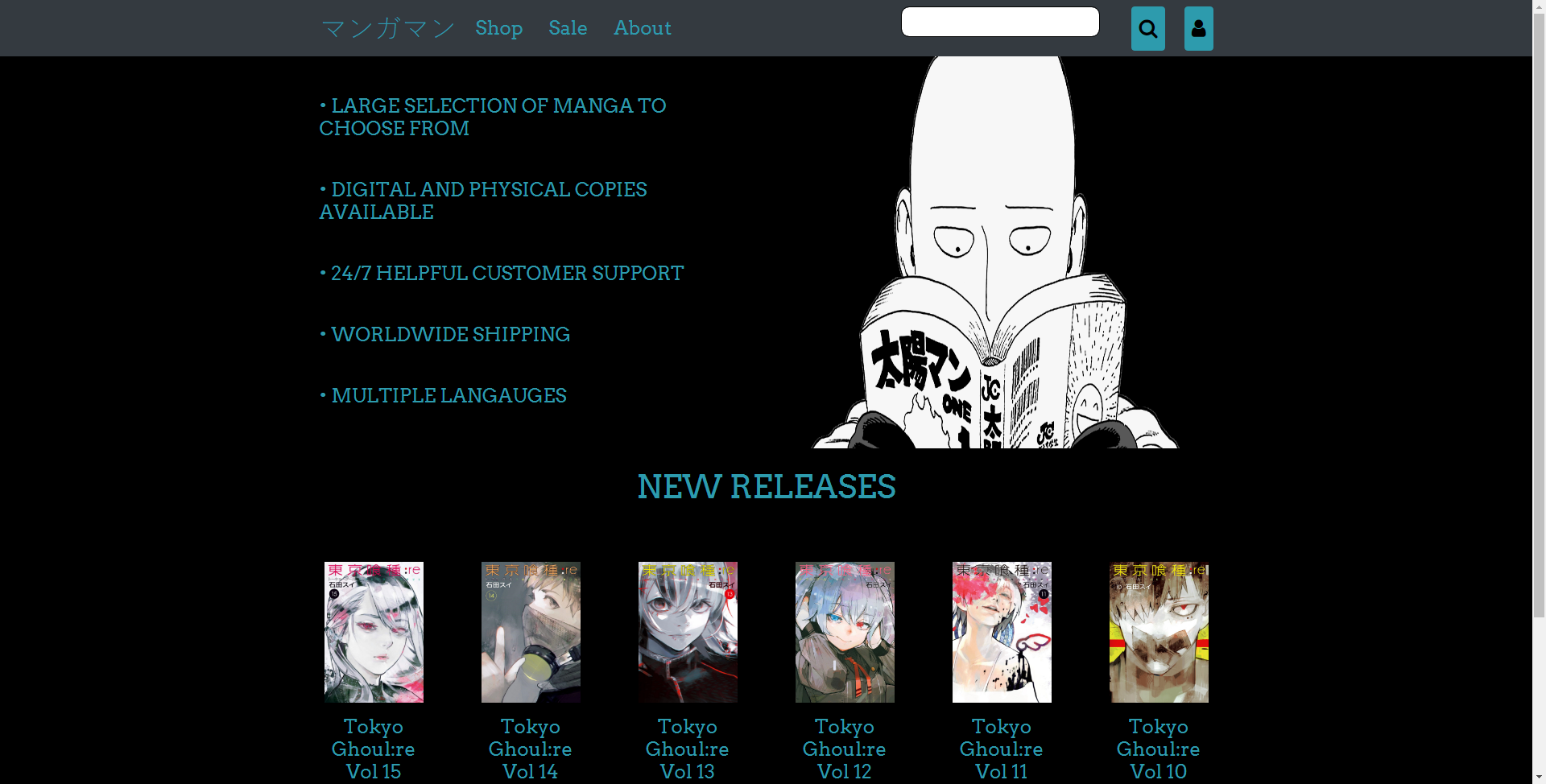
kasutaja lehe vaade:



Külaline

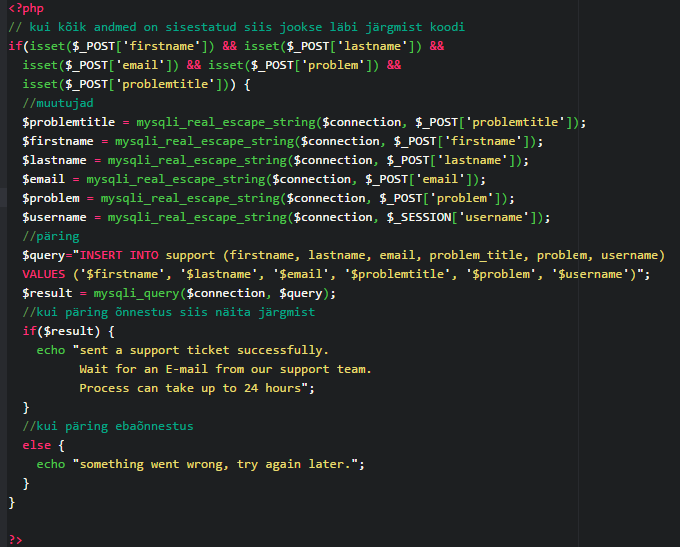
Külalise vaade on väga väike, külaline ei saa midagi osta ega kontakti lehe näha, kuna me ei taha spämmi saada, peab iga külaline registreerima selleks, et midagi teha. Külaline saab vaadata natukene ringi ning kasutada otsingut.

Külalise vaade:



Kontakti lehekülg

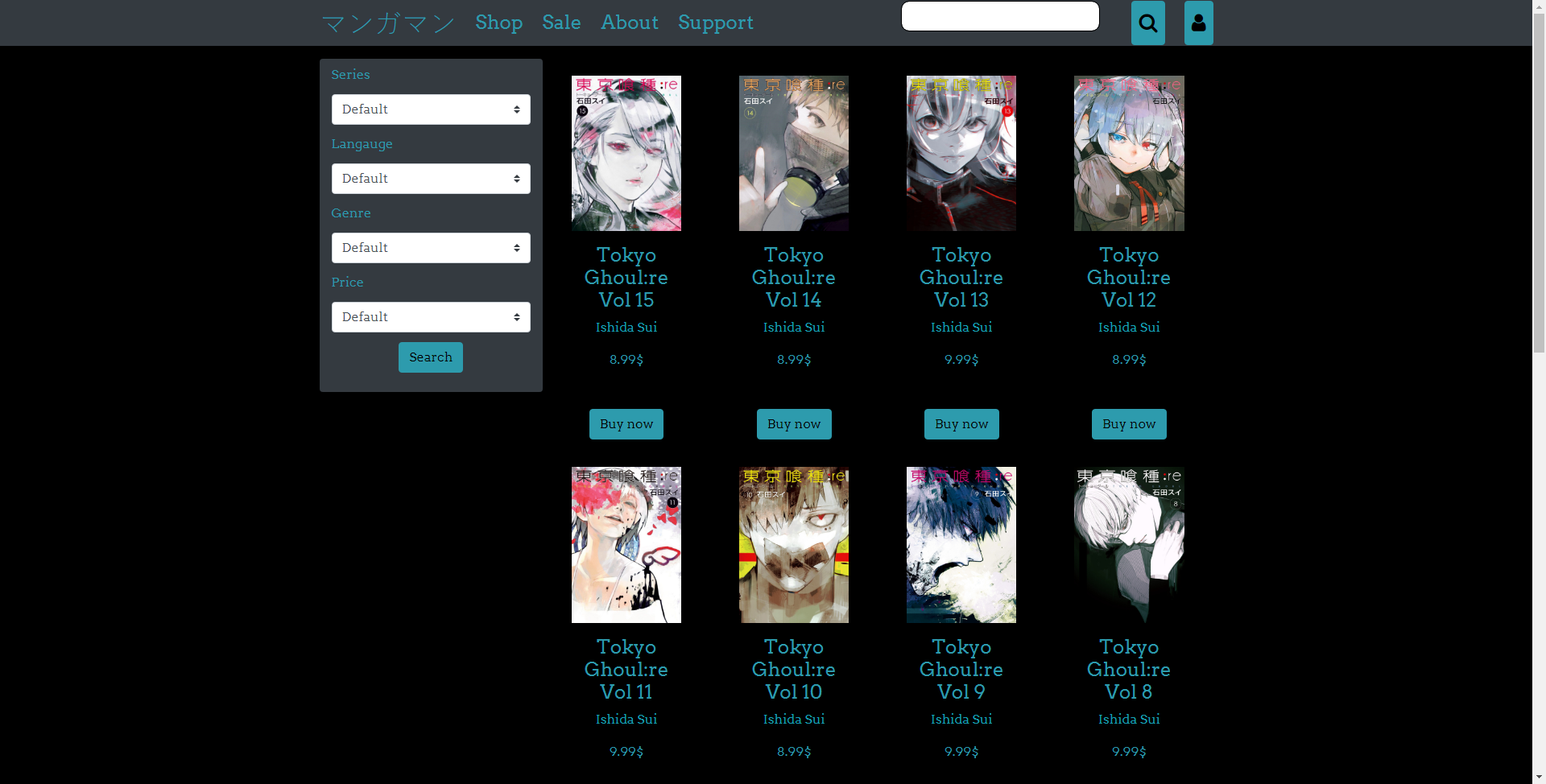
Kontakti lehekülg on mõeldud kasutajatele, kellel on soov ühendust võtta administraatoriga mingi probleemi põhjal. Kasutasime väga lihtsa andmebaasi päringu andemete andmebaasi sisestamisel. Vormi osad mida kasutaja peab sisestama on: Problem Title, First Name, Last Name, E-mail ja Problem description. Kui kõik on sisestatud ning E-mail on valideeritud, sisestatakse kõik andmed andmebaasi järgmise funktsiooniga:



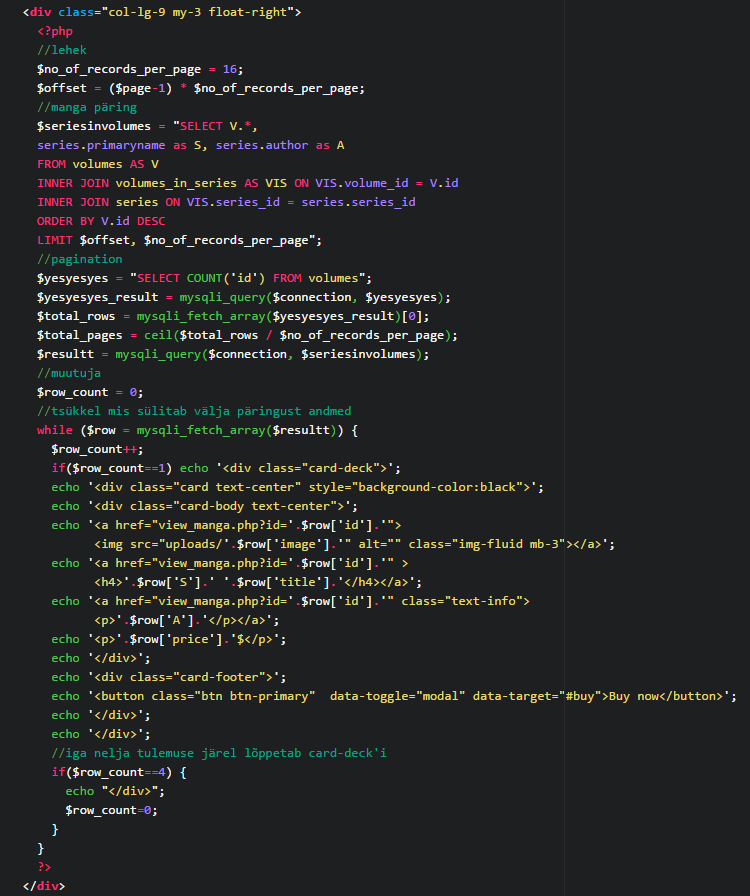
Poe lehekülg

Poe leheküljel on väga palju päringuid ning PHP koodi, kuna kõik andmed sattuvad sinna andmebaasist peab seal mittu andmebaasi ühendamis päringut olema. Poe leheküljel on ka mitu filtreerimise valikut, mis aitavad otsida kasutajat huvitavad mangad, näiteks on võimalik otsida mangade osad sarja, keele, žanri või hinna abil.

Poe lehekülje vaade:



Väike koodi näide:



Testimine

Veebilehe testimine oli pikaldane protsess ning toimus koodimise kõrvalt samaaegselt. Koodi katsetamist tegime lokaalselt ning selleks valitud vahendid olid Google Chrome ja XAMPP. Pärast iga koodijupi kirjutamist toimus selle testimine ning vigade parandamine.

Lõplik testimise eesmärk oli administraatori paneeli funktsioneerimine ning leheküljede töötamine.

**KASUTATUD KIRJANDUS**

1. Info PHP kohta. PHP. [WWW]

<http://et.wikipedia.org/wiki/PHP> (04.10.18)

1. Info MySQL kohta.MySQL.[WWW]

<https://et.wikipedia.org/wiki/MySQL> (04.10.18)

1. Info URL kohta.URL.[WWW]

<https://et.wikipedia.org/wiki/Internetiaadress> (04.10.18)

1. Raamastiku võrdlus.[WWW]

<https://hackernoon.com/top-5-most-popular-css-frameworks-that-you-should-pay-attention-to-in-2017-344a8b67fba1> (04.11.18