

**ТЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ
към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

ДИПЛОМНА РАБОТА

Тема: Онлайн платформа, социален каталог,
с насочност български театри

Дипломант:

Веселина Колова

Научен ръководител:

маг. инж. Кирил Митов

С О Ф И Я

2 0 1 6

УВОД

В днешното забързано ежедневие технологиите набират все повече и повече скорост в развитието си, превръщат се в главните източници на информация и голяма част от вниманието на хората е насочено към тях. А във време, когато общуването лице в лице е на път да изчезне, е важно да се изградят условия това да не се случи и културния живот на човек да се обогатява чрез различни мероприятия.

Пример за такова културно мероприятие е театърът. Той е най-проспериращото българско изкуство.^[1] Количествено българският театър е един от най-продуктивните в Европа, още повече ако се има предвид броят на населението.

Театърът е човешки. Той е мястото, където човек отива, за да слуша. Там е задължен да седи и да чуе какво някой има да каже. Там трябва да се обърне внимание на нещо, което е извън собствения вътрешен монолог. Търси се човека, другия. И най-хубавото е, че така човек намира себе си. Поради това и нуждата от софтуерни приложения, които помагат за намирането на информация за всяка пиеса, от която е заинтересуван и помагат в свързването с други хора, които да дадат оценка и коментар за нея.

До момента са създадени няколко такива платформи, но почти всички са подходящи само за използване на компютър и не поддържат мобилни версии за така наложилите се вече смартфони. Почти всички имат твърде отежнен интерфейс и правят работата с тях трудна за потребителите.

От тази необходимост са породени и целите на настоящата дипломна работа:

- Да се създаде онлайн платформа от тип социален каталог с насоченост български театри;
- Да има всички необходими средства за лесна навигация по актьори, пиеси и театри;

- Опростен дизайн;
- Статистики и търсене;
- Управление на процеса по въвеждане на данни;
- Управление на регистрация и включване към събития.

ГЛАВА I

ПРОУЧВАНЕ НА ПОДОБНИ ОНЛАЙН ПЛАТФОРМИ. ТЕХНОЛОГИИ ЗА ТЯХНАТА РЕАЛИЗАЦИЯ

1.1. Социалните каталози в световен мащаб

1.1.1. Goodreads

Goodreads е уеб сайт за социално каталогизиране на информация, свързана с книги, автори, издателства, цитати от книги, рецензии и отзиви за книги. Създаден е през януари 2007 г. от Отис и Елизабет Чандлър. Позволява на потребителите си свободно да търсят широката, попълвана от потребителите база данни от книги, анотации и рецензии.



Фигура 1.1.1.1. Начална страница на Goodreads

Потребителите на Goodreads могат да се регистрират с профил в системата и да вписват книги, от които да генерират библиотечни каталози и

списъци с книги за четене. Могат и да създават свои групи по интереси и за обсъждания.

През юли 2013 г. бе обявено, че потребителите са надхвърлили 20 милиона.^[2]

1.1.2. Internet Movie Database

Internet Movie Database е онлайн база от данни с информация за актьори, филми, телевизионни предавания, телевизионни звезди, видеоигри и технически персонал. Платформата е стартирана на 17 октомври 1990 г. Става собственост на Amazon.com през 1998 г.

IMDB е най-големият в света уебсайт с информация за филми, като освен за филми, в него има информация също и за актьори, филмографията на актьорите, най-популярните филми, календар на премиерите по години.

Уебсайтът предлага и своя платена версия, наречена IMDB PRO, с която срещу заплащане потребителят може да получи достъп до още по-подробна информация за над 20 000 филма, които са в ранен етап на създаване и за които няма информация никъде другаде.^[3]

1.1.3. Last.fm

Last.fm е музикален уеб сайт, открит през 2002г. Ползвайки системата за препоръки „Audioscrobbler”, Last.fm построява детайлен профил на музикалния вкус на всеки потребител като записва какво слуша от интернет радио станции, от собствения си компютър или от многото преносими музикални устройства. Информацията се прехвърля към базата данни на Last.fm или от самия музикален плейър или от допълнително инсталирано разширение на устройството. След това информацията се показва в профила на потребителя и обработва, за да създаде препратки към страници на индивидуални изпълнители.

Сайтът предлага множество функции на социална мрежа и може да препоръчва и изпълнява изпълнители подобни на любимите на потребителя. Също така включва система, аналогична на тази на Wikipedia, където регистрираните потребители могат да допринесат с информация относно песни, албуми, изпълнители и др.^[4]

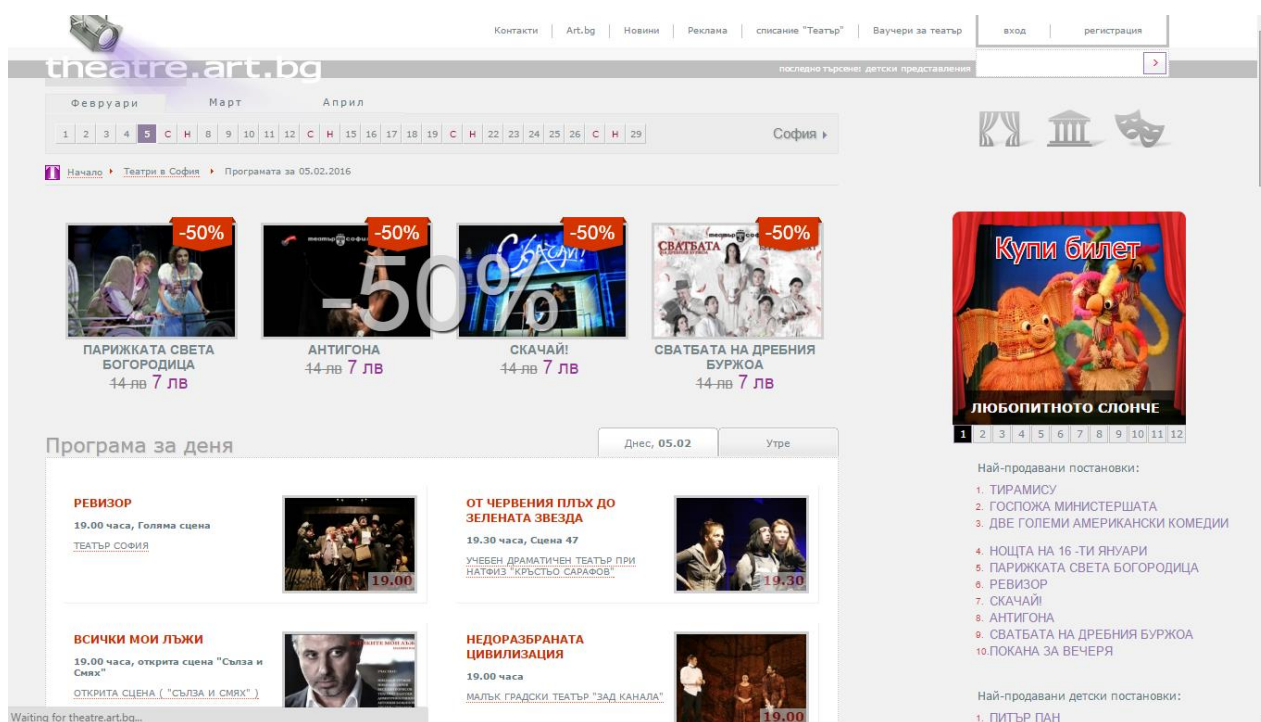
1.2. Български платформи насочени към театралната сцена

1.2.1. Theatre.art.bg

Theatre.art.bg е българска платформа създадена през 1997г., която улеснява информираността на обикновения човек относно предстоящите постановки.

Сайтът дава на потребителите възможността да виждат програмата на театрите в града, кога и къде точно се изпълняват постановките, да закупи билет, да покани приятел, както и да напише мнение за вече гледаното представление или да го оцени.

Уеб сайтът приема разплащания чрез ePay, с банкови карти Visa, MasterCard издадени от български и чужди банки, както и от дебитни карти Маестро и Виза Електрон.^[5]



Фигура 1.2.1.1. Начална страница на theatre.art.bg

1.2.2. Teatri.bg

Teatri.bg е също е насочен към българските театри и е най-младият от разглежданите платформи, създаден през 2007г.

Сайтът запознава потребителя с театрални новини, артисти, предстоящи и минали пиеси, техните рейтинги и рецензии, категоризирани по градове, дати и театри.

Не поддържа закупуване на билети.

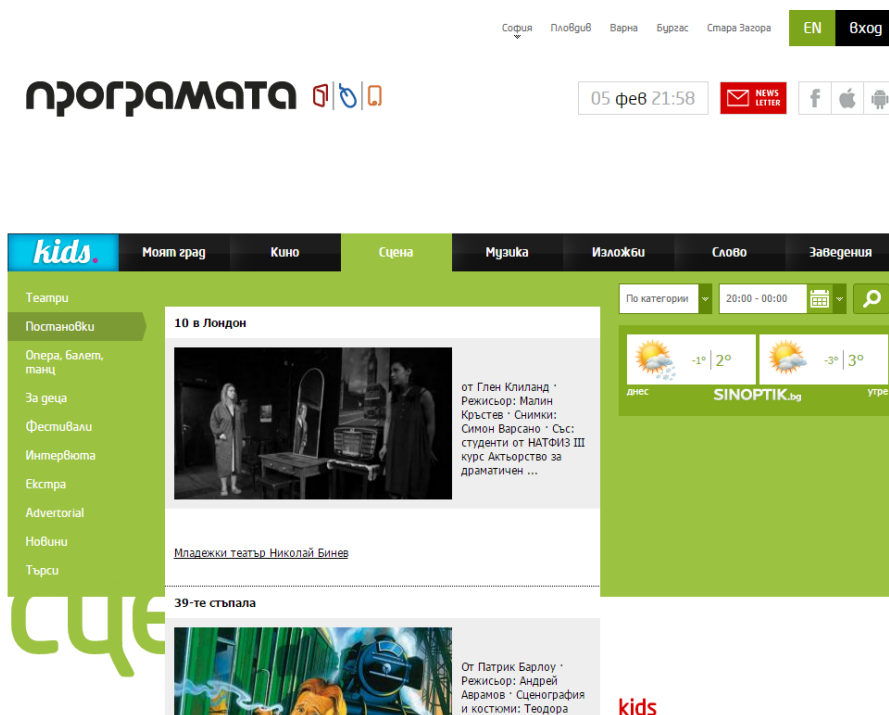


Фигура 1.2.2.1. Начална страница на teatri.bg

1.2.3. Програмата

Програмата е портал, създаден през 2002г., който, за разлика от останалите представени сайтове, показва програмата на събитията от културния живот на българина, като дава информация за множество мероприятия от различни категории.

Регистрираните потребители имат възможност да коментират статиите, новините, местата и събитията, публикувани в уебсайта. Можете да добавяте събития в личния си календар. Можете и да гласувате за заведенията.^[6]

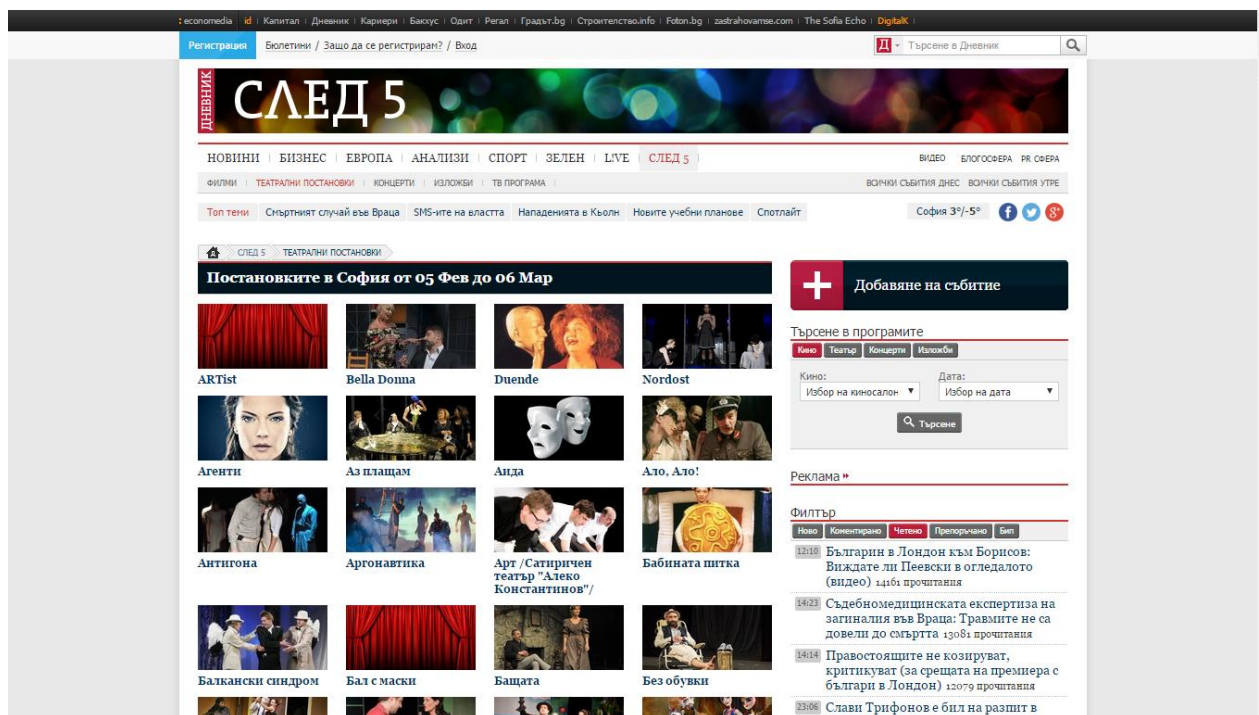


Фигура 1.2.3.1. Начална страница на Програмата

1.2.4. След5

След5 е платформа, която също не е насочена само към театралната сцена. Сайтът дава информация за предстоящи филми, постановки, концерти, изложби, както и за телевизионната програма и е създаден през 2001г.^[7]

Показва се обаче само част от информацията, като се изисква потребителя да е регистриран и влязъл в профила си, за да може да види коментарите и цялостната информация за събитията.



Фигура 1.2.4.1. Начална страница на След5

1.3. Развойни средства и среди

Няколко са съществуващите начини за създаване на уеб сайт, който в основата си се състои от база данни за съхранение на информацията, потребителски интерфейс, служещ за връзката на приложението с потребителите и език за програмиране, с помощта на който се прехвърлят данните между базата данни и интерфейса и се извършват определени дейности в зависимост от функционалността на сайта. Могат да се добавят и различни прикрепени модули към основните компоненти.

Изборът на база данни при разработката на сайт е широк. В зависимост от средствата, с които се разполага, се определя дали да се използва безплатна (MySQL, Firebird, PostgreSQL) или платена (Oracle, MSSQL) база данни. Същото важи и за технологиите за разработка. Уеб сайт може да се създаде с помощта на C#, Perl, Python, Silverlight, Java, PHP и други. Най-разпространените технологии, които се използват са тези на Microsoft .NET, Java и PHP. За продуктите на Microsoft обикновено се заплаща, при което за разработката на приложенията е необходимо наличието на Visual Studio. Java

се използва бесплатно. PHP е най-използваният език за създаване на уеб сайтове.^[8] За разработката на продукт с тази технология не се заплаща, а редакторите и средите за разработка, които могат да се ползват са доста. За направата на сайт чрез PHP може да се започне от нулата, но е възможно и да се използва софтуерна рамка (framework), която да имплементира основата на уеб приложението, като така да се започне работа само по функционалността и интерфейса, или система за управление на съдържание.

За да се започне един проект „от нулата” обаче се изискват много усилия и могат да се допуснат доста грешки при разработката и като резултат да се получи непълен и/или неработещ продукт. При функционалното програмиране се използват предимно функции, на които се предават аргументи, което довежда до затруднение от страна на разработчика да следи какви аргументи къде се подават. При обектно-ориентираното програмиране се използват обекти и нещата са по-улеснени, прегледни и логични, но отново, ако се започне работата върху проект „от нулата” се изисква много време и внимание. Затова са създадени софтуерните рамки, които рационализират създаването на уеб приложенията, като предоставят базова структура на програмиста, върху която той да изгради даден продукт. Това помага да се спести време и да се разработват по-стабилни приложения, като се отделя повече време на функционалността и детайлите на сайта и се намалява писането на повтарящ се код.

Третият начин за създаването на уеб сайт е с помощта на т.н. системи за управление на съдържание (Content Management System (CMS)).^[9] Те имат за цел да улеснят изграждането на динамичен уеб сайт, с възможности за лесна промяна в съдържанието му по всяко време. Главното, с което една такава система е привлекателна, е лекотата на употреба. Затова този тип софтуер за добър за всеки, който има екип хора, занимаващи се с поддръжката на сайт, които нямат умения в областта на уеб разработката или имат много малко технически опит. Идеята на този вид технологии е да се избегне или поне да се

намали нуждата от ръчно писане на код. Това намалява необходимостта от специалисти, които се занимават с попълването на информация в уебсайтовете.

Основните характеристики на една завършена система за управление на съдържанието са следните:

- лесно и бързо публикуване на документи и материали;
- идентификация на потребителите и разпределянето им в групи с различни права за достъп;
- възможността да се дава определено ниво на достъп до категории от съдържанието или до отделни документи в него;
- съхранение онлайн на всякакви видове материали (снимки, видео, документи, програми и др.)

В момента съществуват множество CMS системи на Java, Perl, PHP, Python, Ruby on Rails, JavaScript и др.

ГЛАВА II

ИЗИСКВАНИЯ КЪМ ПРОГРАМНИЯ ПРОДУКТ. ИЗБОР НА РАЗВОЙНА СРЕДА И СРЕДСТВА

2.1. Изисквания

Платформата трябва да предоставя следните възможности:

- средства за навигация по актьори, събития, театри;
- управление на процеса по въвеждане на данни;
- статистики и търсене върху актьори, събития, театри;
- управление на регистрация и включване към събития.

2.2. Избор на програмни езици

2.2.1. PHP

PHP е широко използван скриптов език с отворен код, предназначен за обща употреба, който е изключително удобен за уеб разработки и може да се вгражда в HTML.

Това, което отличава PHP от примерно клиентски JavaScript, е че кодът се изпълнява на сървъра и генерира HTML, който се изпраща към клиента. Клиентът ще получи резултата от изпълнението на този скрипт, но няма да знае какъв код е генерирал този резултат. Дори можете да конфигурирате уеб сървъра си да обработва всички HTML-файлове с PHP и тогава няма как потребителите да знаят какво правите зад гърба им.^[10]

Съществуват множество модули (разширения) за PHP, които добавят различни функционалности и позволяват много по-бързо и ефективно разработване. Такива са:

- функции за работа с низове и регулярни изрази;

- функции за дата и час;
- математически функции;
- функции за управление на сесии и работа с бисквитки (cookies);
- функции за SOAP;
- функции за работа с различни бази данни;
- функции за изпращане на e-mail съобщения;
- и др.

PHP може да работи с повечето модерни бази данни – MySQL, PostgreSQL, Microsoft SQL Server, Oracle, SQLite и др.

Най-големите достойнства на PHP при реализирането на уеб проекти:

- бързо;
- евтино;
- богат избор от изпълнители;
- богат избор от хостинг компании.

2.2.2. JavaScript

JavaScript е скриптов език създаден от екипа на Netscape преди малко повече от 20 години, ползван при развиването на динамични интернет страници. Разпространява се с повечето браузъри. Поддържа както обектно-ориентирано, така и функционално програмиране.

Поради голямата си популярност в интернет пространството съществуват множество библиотеки и разработки.

Най-ползваните възможности на едно JavaScript приложение са:

- Получаване на съдържанието на страница и изпращане на данни към сървър чрез AJAX заявки без да има нужда от презареждане на страницата;

- Различни ефекти, приложени върху HTML елементи (скриване/показване, пренареждане, влачене, слайд шоу, анимация, филтри върху изображенията и др.);
- Интерактивно съдържание;
- Валидация на входни данни от уеб форма;
- Управление над прозорци и рамки;
- Разпознаване на възможностите на браузъра;
- Раздвижен потребителски интерфейс.

2.3. Сравнение между WordPress, Joomla и Drupal

WordPress, Joomla и Drupal са се установили като лидери сред системите за управление на съдържание.^[11] Те са трите най-известни и имат няколко общи неща по между си:

- WordPress, Joomla и Drupal са безплатни и с лиценз за отворен код;
- И трите са написани предимно на PHP;
- Поддържат MySQL като тяхна система за управление на бази данни, макар че за WordPress тя е единствена, докато Joomla и Drupal поддържат и други;
- И трите ползват теми и шаблони за визуално представяне на сайтове, разширения;
- Като софтуер с отворен код, и трите се движат и развиват от обществото потребители.



WordPress

Пусната за разпространение през 2003 г. от Мат Муленвег

Joomla

Пусната за разпространение през 2005 г., разклонение на Mambo

Drupal

Пусната през 2001 г. от Дрис Бейтарт

ПОПУЛЯРНОСТ		
> 160 милиона изтегляния	> 50 милиона изтегляния	> 15 милиона изтегляния
ЦЕНА		
безплатна	безплатна	безплатна
БЕЗПЛАТНИ ТЕМИ		
3 000+	1 000+	2 000+
БЕЗПЛАТНИ ПЛЪГИНИ		
36 000+	7 000+	26 000+
ИНСТАЛАЦИЯ С ЕДНО КЛИКВАНЕ		
✓	✓	✓
ВРЕМЕ ЗА РЪЧНА ИНСТАЛАЦИЯ		
5 минути	10 минути	10 минути
НЕОБХОДИМО НИВО НА ТЕХНИЧЕСКИ УМЕНИЯ		
★	★★	★★★
ЧЕСТОТА НА ОБНОВЛЕНИЯТА		
42 дни	36 дни	51 дни

Таблица 1. Бързо сравнение на WordPress, Joomla и Drupal^[12]

2.3.1. WordPress

WordPress е най-известната система за управление на съдържание.^[13] Стартира като платформа за блогове, но през годините се е развила значително

и вече е приложима при разработката на сложни новинарски ресурси и дори интернет-магазини. Днес над 60 милиона уеб сайтове използват WordPress.

Системата предлага много предимства за тези, които искат да създадат сайт или блог, включително:

- Лесна инсталация – много компании за хостинг предлагат автоматична инсталация на WordPress сайтове, което значи че можете да имате работещ сайт за по-малко от 5 минути (или до час ако нямате никакъв опит и го инсталирате ръчно);
- Персонализиране – WordPress има значително повече разширения, теми и други персонализации от всяка друга CMS, които потребителят може да изтегли и инсталира директно от административния панел;
- Най-добре оптимизираната система за видимост от интернет търсачки;
- Голям набор от информация – несъмнено поради популярността си.

WordPress обаче не е перфектна. Има чести оплаквания, че ако сайтът стане твърде голям, зареждането му може да стане по-трудно. Системата също така е трудна за промяна и тези, които искат да направят някои промени по функционалността ѝ, може да се сблъскат със сериозни проблеми.

2.3.2. Drupal

Drupal е втората най-използвана CMS днес. Платформата ѝ е изключително мощна и използва по-малко ресурси от тази на WordPress. Drupal може да се ползва за създаване както на малък блог, така и на портал със съдържание на големи корпорации. Някои от най-значимите предимства на Drupal включват:

- Техническа напредналост – Drupal е най-напредналата технически от трите системи. Не ползва толкова много ресурси като WordPress, затова хората няма нужда да се тревожат за преместване на по-скъп хостинг бързо;
- Подобрена работа – страниците на Drupal по принцип се зареждат по-бързо и имат по-бързо време за отговор от тези на WordPress и Joomla.

Разбра се, ако добавяте разширения и правите други промени, това може да се промени.

- Персонализация – Drupal е лесен за персонализиране с много различни разширения, теми и други опции за конфигурация. Възможно е да се редактират дори файловете на програмата в ядрото ѝ, което я прави най-гъвкавата от трите системи.

Drupal се счита за най-мощната CMS, но с тази сила се появяват допълнителни трудности за собственика на сайта. Силно се препоръчва всеки, който обмисля използването на Drupal, да има поне базови знания от HTML, PHP и други често срещани програмни езици в уеб пространството. Въпреки че Drupal предполага по-голяма гъвкавост, администрацията е доста сложна за програмиста. Клиентите трябва да наемат експерт в Drupal и общите потребители на сайта не могат лесно да бъдат обучени да използват Drupal, както могат да бъдат обучени, за да използват Joomla.^[14]

2.3.3. Joomla

Сравнявайки трите най-известни системи за управление на съдържание с отворен код, Joomla често е виждана до WordPress и Drupal като равностойна, но различна CMS опция. Системата е мощна и може да работи на повечето уеб сървъри без проблеми. Не изисква същото ниво на технически знания като Drupal, но продължава да предлага множество от допълнителните функции. Ако управлявате сайт, който е тежък откъм съдържание, административният панел ще го понесе напълно.

Joomla няма красивия интуитивен вид, който WordPress има, нито възможностите за персонализация на Drupal, а е някъде по средата. Други причини, поради които хората избират Joomla включват:

- Социални мрежи – вероятно най-голямото предимство на Joomla. От трите разгледани, тя прави създаването на социални мрежи най-лесно;

- Онлайн магазини – създаването, на които също става много лесно. Докато е възможно с WordPress и Drupal, то Joomla има по-добра поддръжка за това в ядрото си;
- Помощен портал – Joomla предлага страхотен помощен портал за задаване на въпроси и получаване на техническа подкрепа. Не е толкова бърз и обширен като този на обществото потребители на WordPress, но е по-бърз (и по-евтин) от техническата поддръжка, за която повечето хора, ползващи Drupal, плащат.

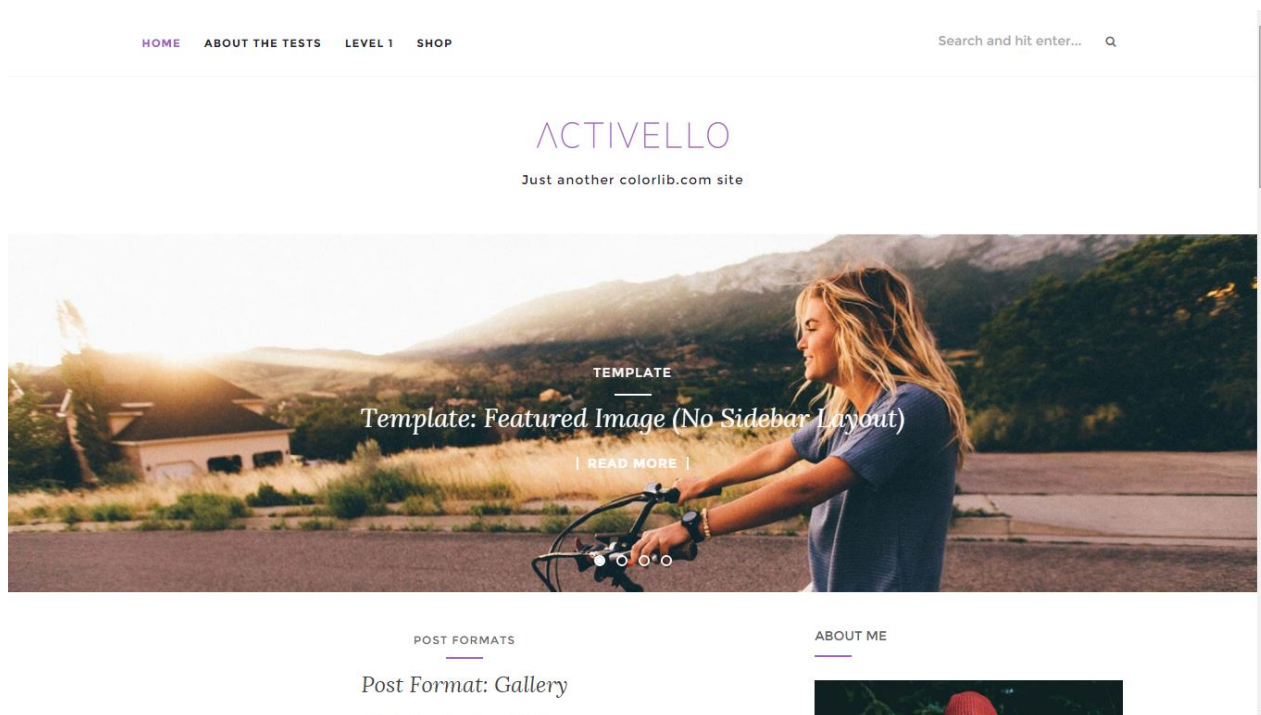
Много потребители харесват Joomla, защото е мощна и едновременно с това лесна за употреба. Системата е свършила добра работа с комбинирането на предимствата на WordPress и Drupal, като е вмъкнала и някои свои услуги. Последните няколко години популярността ѝ нараства и вероятно ще продължи да го прави.

Въз основа на сравнението на трите системи избраната за осъществяване целите на дипломната работа е WordPress.

2.4. Допълнителни средства

2.4.1. Activello

Activello е изчистена, универсална WordPress тема, ползваща Bootstrap. Разполага с прекрасни графики и дизайн. Удобна е за употреба както на компютри, така и на мобилни устройства.



Фигура 2.4.1.1. Начална страница на Activello

Темата разполага с адаптивен дизайн, подходяща е за Retina дисплей, поддържа WordPress персонализация, има включен слайдър¹, поддържа повечето WordPress разширения, включително и WooCommerce, преведена е на няколко езика, има хубави картинки и неограничени опции относно цветовете.^[15]

2.4.2. WooCommerce


WooCommerce е безплатно разширение за онлайн търговия, което позволява да се продават както физически, така и електронни продукти. Създадено е да се интегрира безпроблемно с WordPress и дава пълен контрол както на собствениците на магазина, така и на разработчиците.^[16]

Първоначално бе избран подобното разширение за онлайн продажби на impleCode, но след възникнали технически проблеми (продуктът спираше

¹ В дизайн терминологията така се нарича слайд шоу поставено в уеб страница.

работа след изключване и включване на хоста, въпреки помощта от страна на разработчиците) се наложи замяна с WooCommerce.

[Home/ Shop/ Clothing/ Hoodies/ Happy Ninja](#)



Happy Ninja
★★★★☆ (2 customer reviews)

£10.00–£12.00

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

Color Red
[Clear selection](#)

£10.00

1

SKU: N/A. Categories: [Clothing](#), [Hoodies](#).

Description **Discussion (5)**

Product Description

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

Фигура 2.4.2.1. Примерна страница на продукт в WooCommerce

В момента WooCommerce е зад 30% от онлайн магазините – повече от всяка друга платформа.^[17]

Разширението е с отворен код.

2.4.3. BuddyPress

BuddyPress е мощно разширение за WordPress, което превръща сайта в напълно работеща социална мрежа.^[18]

BuddyPress е създаден от няколко по-малки разширения.

Всяко разширение се ползва, за да добави социална функционалност към сайта и може да работи независимо от останалите, така че могат да се изберат всички или само отделно разширение.

Тези разширения са:

- **Разширени профили** – разрешава да се създават допълнителни полета в потребителския профил;
- **Лични съобщения** – разрешава на потребителите да общуват помежду си директно;
- **Приатели** – потребителите могат да го ползват, за да създават лист от приятели, както във Facebook или Ning;
- **Групи** – разрешава на потребителите да създават групи, където да споделят информация по теми, които ги интересуват;
- **Activity Streams** – записи за всяка активност за всеки потребител, както и цялата активност на всички потребители в цялата мрежа. Подобно на „стената“ на Facebook;
- **Следене на блог** – следи сайтове в мрежата включително наскоро създадени, последни постове и последни коментари;
- **Форуми** – позволява на групи да създават и управляват собствени форуми.

2.4.4. jQuery

jQuery е най-известната и използвана библиотека за JavaScript.^[19] Създава я Джон Резиг и я пуска официално в Интернет през 2006 година.

Библиотеката е проектирана, за да улесни и опрости писането на JavaScript код, като едновременно с това е гъвкава и може да се надгражда с допълнителни разширения.^[20]

2.4.5. jQuery UI

jQuery User Interface е помощен пакет с ефекти, джаджи и теми построени върху jQuery библиотеката, влияещи на потребителския интерфейс.

2.4.6. Bootstrap

Bootstrap, първоначално наречен Twitter Blueprint е безплатна софтуерна рамка, комбинация от HTML, CSS и JavaScript, създадена от екипа на Twitter.

Платформата се харесва на разработчиците поради следните си характеристики:

- Скорост;
- Лека, елегантна и приспособима;
- Обширен списък на компонентите;
- Пакети с JavaScript разширения;
- Добра документация.

Bootstrap е най-отбелязваният със звезда проект в GitHub с над 90 000 звезди.^[21]

2.5. Структура на базата данни

2.5.1. MySQL

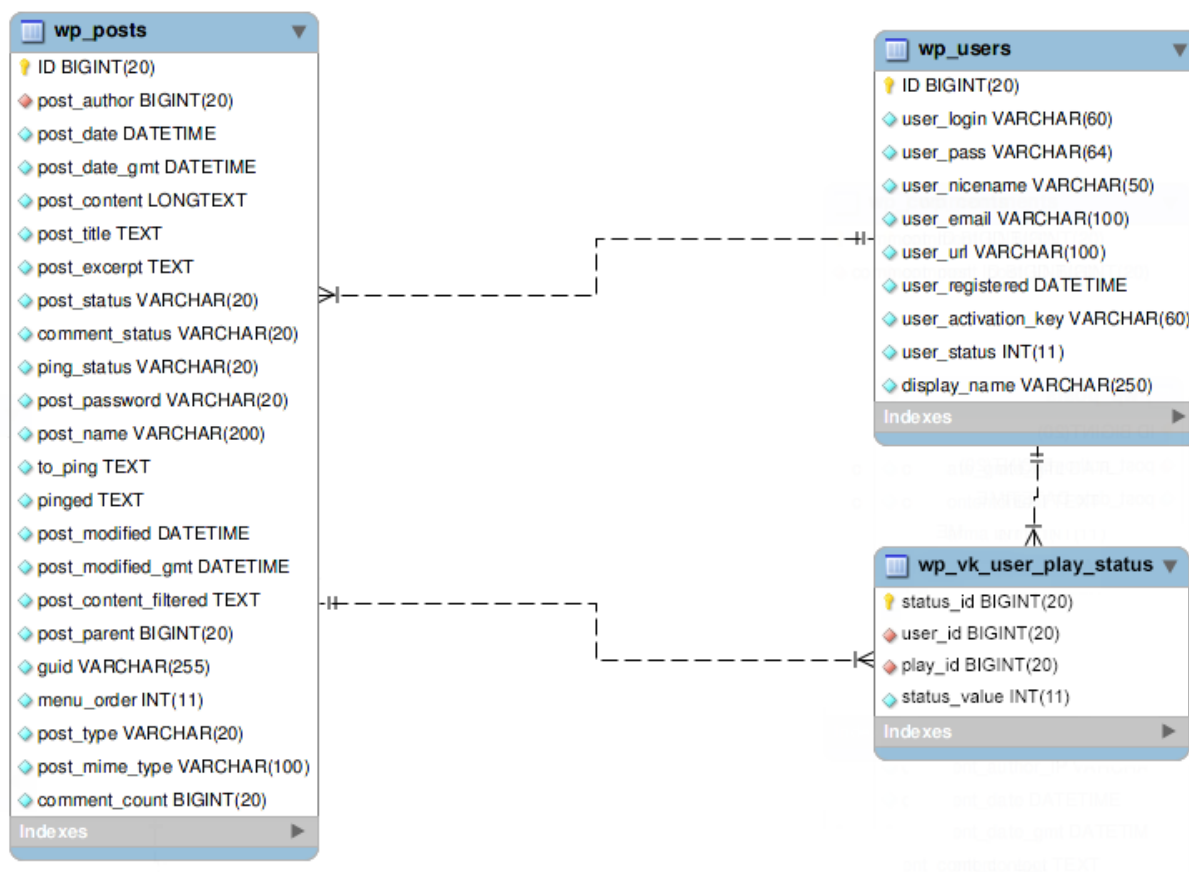
MySQL е SQL система за управление на бази данни, която е предназначена да работи с големи натоварвания и да осъществява сложни заявки. MySQL е многонишков и многопотребителски сървър. Използва се SQL, който е стандартен световно разпространен език за заявки към база данни. Като система за релационни бази данни MySQL позволява свързването на множество и различни таблици за максимални ефективност и скорост.

Бързата многонишкова система за заделяне на памет позволява след изпълнение на заявката тя веднага да се освобождава. Това е от голямо значение при продължителна работа и при много на брой свързани клиенти.

MySQL поддържа клиентски API (приложни интерфейси) за широк набор от програмни езици като Perl, C, PHP и др. Клиентските програми, извършващи достъп до данните в MySQL база данни могат да се напишат с помощта на тези API. В това отношение MySQL предоставя голям избор за програмен език при реализирането на среден слой.

MySQL се разпространява с лиценз за отворен код.

2.5.2. Структура на базата



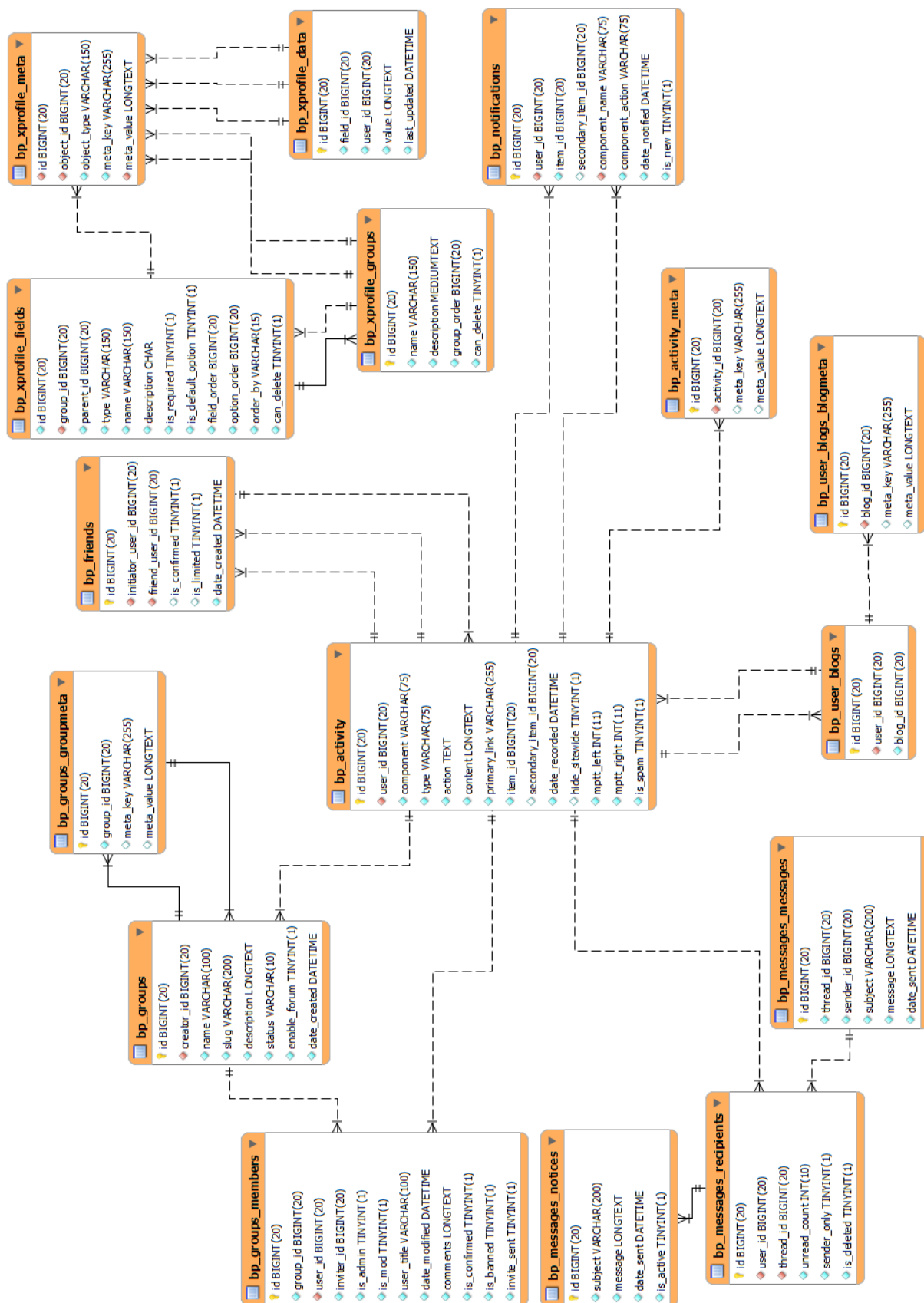
Фигура 2.5.2.1. Реляционна схема

На фигура 2.5.2.1. е показана малка част от релационната схема на платформата. Добавена е таблицата “vk_user_play_status”, в която се съхраняват записите от това като каква е отбелязал потребителят дадена постановка.

Поleta на „vk_user_play_status“:

- **status_id** – идентификационен номер;
- **user_id** – идентификационен номер на потребителя;
- **play_id** – идентификационен номер на постановката;
- **status_value** – стойност, указваща дали потребителят е гледал пиесата или иска да гледа.

На следващата фигура са показани таблиците на разширението BuddyPress и техните връзки.



Фигура 2.5.2.1. Релационна схема на BuddyPress

ГЛАВА III

ПРОГРАМНА РЕАЛИЗАЦИЯ НА ОНЛАЙН ПЛАТФОРМА, СОЦИАЛЕН КАТАЛОГ С НАСОЧЕНОСТ БЪЛГАРСКИ ТЕАТРИ

3.1. Управление на процеса по въвеждане на данни

Благодарение на CMS-а WordPress и разширението му WooCommerce, това е наполовина свършено. WooCommerce има своя персонализирана публикация (Custom Post Type)^{[22][23]}, която почти изцяло покрива нуждите ни. Има поле за заглавие, описание, етикети, категории, атрибути, изображение, галерия. Разширението има и други полета за информация, но споменатите току-що са тези от които се нуждаем, така че може да премахнем останалото.

Необходими са обаче и полета за допълнителна информация – кога, от колко часа, на каква сцена ще се играе представлението и от къде (ако е възможно) да се закупят билети. Затова е необходимо добавянето на още една мета кутия (meta box).

```
1. add_meta_box( 'vk_details',  
    'Детайли за пиесата', 'vk_details_output', 'product' );
```

Полетата се извеждат в мета кутията:

```
1. function vk_details_output($post) {  
2.     wp_nonce_field(basename(__FILE__), "meta-box-nonce");  
3.     $meta_dates = get_post_meta( $post->ID, '_vk_date_input', true );  
4.     $meta_time = get_post_meta( $post->ID, '_vk_time_input', true );  
5.     $meta_scene = get_post_meta( $post->ID, '_vk_scene_input', true );  
6.     $meta_tickets = get_post_meta( $post->ID, '_vk_tickets_input', true );  
7.     ?>  
8.     <p>Датите за пиесата във формат ДД/ММ/ГГГГ, разделени със запетайки:  
    <input type="text" name = "vk_date_input" id="vk_date_input" value="<?php  
    echo $meta_dates;?>" size="30" /></p>  
9.     <p>Начален час: <input type="text" name = "vk_time_input"  
    id="vk_time_input" value="<?php echo $meta_time; ?>" size="30" /></p>  
10.    <p>Сцена: <input type="text" name = "vk_scene_input"  
    id="vk_scene_input" value="<?php echo $meta_scene; ?>" size="30" /></p>  
11.    <p>Билети: <input type="url" name = "vk_tickets_input"  
    id="vk_tickets_input" value="<?php echo $meta_tickets; ?>" size="30" /></p>
```

```
12.     <?php
13.     }
```

След публикуване/обновяване на пиесата се запазват:

```
1. add_action( 'save_post', 'vk_info_save', 10, 3 );
2. function vk_info_save( $product_type, $post_id) {
3.     global $post, $thepostid;
4.     $thepostid = $post->ID;
5.
6.     update_post_meta( $post_id, '_vk_date_input', $_POST['vk_date_input']
7. );
8.     update_post_meta( $post_id, '_vk_time_input', $_POST['vk_time_input']
9. );
10.    update_post_meta( $post_id, '_vk_scene_input',
11. $_POST['vk_scene_input'] );
12.    update_post_meta( $post_id, '_vk_tickets_input',
13. $_POST['vk_tickets_input'] );
14.
15.    if (!isset($_POST["meta-box-nonce"]) ||
16. !wp_verify_nonce($_POST["meta-box-nonce"], basename(__FILE__)))
17.        return $post_id;
18.
19.    $meta_box_date_value = "";
20.    $meta_box_time_value = "";
21.    $meta_box_scene_value = "";
22.    $meta_box_tickets_value = "";
23.
24.    if(isset($_POST["vk_date_input"])) {
25.        $meta_box_text_value = $_POST["vk_date_input"];
26.    }
27.    update_post_meta($thepostid, "_vk_date_input",
28. $meta_box_text_value);
29.
30.    if(isset($_POST["vk_time_input"])) {
31.        $meta_box_time_value = $_POST["vk_time_input"];
32.    }
33.    update_post_meta($thepostid, "_vk_time_input",
34. $meta_box_time_value);
35.
36.    if(isset($_POST["vk_scene_input"])) {
37.        $meta_box_scene_value =
38. $_POST["vk_scene_input"];
39.    }
40.    update_post_meta($thepostid, "_vk_scene_input",
41. $meta_box_scene_value);
42.
43.    if(isset($_POST["vk_tickets_input"])) {
44.        $meta_box_tickets_value =
45. $_POST["vk_tickets_input"];
46.    }
47.    update_post_meta($thepostid, "_vk_tickets_input",
48. $meta_box_tickets_value);
49. }
```

3.2. Включване към събития

Към събитие могат да се включат само регистрирани потребители, след като отворят страницата на постановка. Включването като такова представлява грубо казано прост запис в базата с това дали потребителя иска да види или вече е гледал дадената пиеса.

За целта на страницата на постановката първо трябва да се покажат бутоните за избор. Това става като първо се провери дали в базата има съществуващ запис за потребителя и пиесата и според резултата чрез функцията `add_status_buttons()` се извеждат по подходящ начин:

```
1. function add_status_buttons() {
2.     global $product;
3.     do_action('vk_enqueue_scripts');
4.     ?>
5.     <input type="hidden" id="play_id" value="<?php $play_id = absint(
6.         $product->id ); echo $play_id; ?>" />
7.     <input type="hidden" id="user_id" value=" <?php $user_id = the_ID();
8.         echo $user_id; ?> " />
9. <?php
10.     if(is_user_logged_in()) {
11.         global $wpdb;
12.         $result = $wpdb->get_row($wpdb->prepare("SELECT * FROM
13.             {$wpdb->prefix}vk_user_play_status WHERE user_id = " . $user_id . " AND
14.             play_id = " . $play_id));
15.     ?>
16.     <!-- Split button -->
17.     <div class="btn-group">
18.         <button type="button" class="btn btn-<?php
19.             echo ( !count($result) ? "success" : "default" ); ?>"
20.             data-toggle="dropdown" id="<?php echo ( $result->status_value ? "have_seen"
21.                 : "want_to_see"); ?>">
22.             <?php
23.                 echo ( !count($result) ? "" : '<span class="glyphicon
24.                     glyphicon-ok" aria-hidden="true"></span> ' );
25.                 echo ( $result->status_value ? esc_html( $product-
26.                     >mark_as_seen_text() ) : esc_html( $product->mark_as_want_to_see_text() ) );
27.             ?>
28.         </button>
29.         <button type="button" class="btn btn-success dropdown-toggle" data-
30.             toggle="dropdown" aria-haspopup="true" aria-expanded="false">
31.             <span class="caret"></span>
32.             <span class="sr-only">Toggle Dropdown</span>
33.         </button>
34.         <ul class="dropdown-menu">
```

```

25.         <li><a id="<?php echo ( !$result->status_value ? "have_seen" :
    "want_to_see"); ?>" ><?php echo ( !$result->status_value ? esc_html(
    $product->mark_as_seen_text() ) : esc_html( $product-
    >mark_as_want_to_see_text() ) ); ?></a></li>
26.     </ul>
27. </div>
28.     <?php
29. }
30. }

```

Функцията е прикрепена към действието (action) „vk_add_status_buttons“, което се извиква в шаблона на продукта/пиесата след категориите и на мястото на етикетите:

```

1. do_action( 'vk_add_status_buttons' );

```

Натискането на бутоните се обработва от JavaScript, който изпраща ајак заявка до WordPress, за да създаде/обнови/изтрие запис, а бутоните се обновяват според върнатия резултат.

```

1. (function($) {
2.
3.     $( "#have_seen" ).live( "click", function() {
4.         $( "#have_seen" ).empty().append('
    запазване...');
5.
6.         $.ajax({
7.             url: vk_select_params.vk_ajax_url,
8.             data: {
9.                 'action': 'update_seen',
10.                 'security': vk_select_params.ajax_nonce,
11.                 'user_id': $( '#user_id' ).val(),
12.                 'play_id': $( '#play_id' ).val()
13.             },
14.             type: 'POST',
15.             success: function( response ) {
16.                 switch ( response ) {
17.                     case '0':
18.                         $( ".btn-group" ).empty().append( '<button
    type="button" class="btn btn-default" data-toggle="dropdown"
    id="want_to_see">
                                <span class="glyphicon glyphicon-ok"
    aria-hidden="true"></span> Искам да я гледам
                                </button>
                                <button type="button" class="btn btn-success dropdown-toggle" data-
    toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                                <span
    class="caret"></span>
                                <span class="sr-only">Toggle
    Dropdown</span>
                                </button>
                                <ul class="dropdown-menu">
                                <li><a
    id="have_seen">Гледах я</a></li>
                                </ul>' );
19.                     break;

```



```

20.         case '1':
21.             $( ".btn-group" ).empty().append( ' <button
type="button" class="btn btn-default" data-toggle="dropdown"
id="have_seen"> <span class="glyphicon glyphicon-ok"
aria-hidden="true"></span> Гледах я </button> <button
type="button" class="btn btn-success dropdown-toggle" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false"> <span
class="caret"></span> <span class="sr-only">Toggle
Dropdown</span> </button> <ul class="dropdown-menu"> <li><a
id="want_to_see">Искам да я гледам</a></li> </ul>' );
22.             break;
23.         default:
24.             $( ".btn-group" ).empty().append( '<button
type="button" class="btn btn-success" data-toggle="dropdown"
id="want_to_see"> Искам да я
гледам </button> <button type="button" class="btn btn-
success dropdown-toggle" data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false"> <span class="caret"></span> <span
class="sr-only">Toggle Dropdown</span> </button> <ul
class="dropdown-menu"> <li><a id="have_seen">Гледах я</a></li>
</ul>' );
25.         }
26.     }
27.     });
28. });
29.
30. $( "#want_to_see" ).live( "click", function() {
31.     $( "#want_to_see" ).empty().append( '
запазване...' );
32.
33.     $.ajax({
34.         url: vk_select_params.vk_ajax_url,
35.         data: {
36.             'action': 'want_to_see',
37.             'security': vk_select_params.ajax_nonce,
38.             'user_id': $( '#user_id' ).val(),
39.             'play_id': $( '#play_id' ).val()
40.         },
41.         type: 'POST',
42.         success: function( response ) {
43.             switch ( response ) {
44.                 case '0':
45.                     $( ".btn-group" ).empty().append( '<button
type="button" class="btn btn-default" data-toggle="dropdown"
id="want_to_see"> <span class="glyphicon glyphicon-ok"
aria-hidden="true"></span> Искам да я гледам </button>
<button type="button" class="btn btn-success dropdown-toggle" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false"> <span
class="caret"></span> <span class="sr-only">Toggle
Dropdown</span> </button> <ul class="dropdown-menu"> <li><a
id="have_seen">Гледах я</a></li> </ul>' );
46.                     break;
47.                 case '1':

```

```

48.         $( ".btn-group" ).empty().append( ' <button
type="button" class="btn btn-default" data-toggle="dropdown"
id="have_seen"> <span class="glyphicon glyphicon-ok"
aria-hidden="true"></span> Гледах я </button> <button
type="button" class="btn btn-success dropdown-toggle" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false"> <span
class="caret"></span> <span class="sr-only">Toggle
Dropdown</span> </button> <ul class="dropdown-menu"> <li><a
id="want_to_see">Искам да я гледам</a></li> </ul>' );
49.             break;
50.         default:
51.             $( ".btn-group" ).empty().append( '<button type="button"
class="btn btn-success" data-toggle="dropdown"
id="want_to_see"> Искам да я
гледам </button> <button type="button" class="btn btn-
success dropdown-toggle" data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false"> <span class="caret"></span> <span
class="sr-only">Toggle Dropdown</span> </button> <ul
class="dropdown-menu"> <li><a id="have_seen">Гледах я</a></li>
</ul>' );
52.         }
53.     }
54. });
55. });
56. })(jQuery);

```

Обработката на заявките:

```

1. <?php
2.
3. if ( ! defined( 'ABSPATH' ) ) {
4.     exit; // Exit if accessed directly
5. }
6.
7. add_action( 'vk_enqueue_scripts', 'vk_my_enqueue' );
8. function vk_my_enqueue($hook) {
9.     wp_enqueue_script( 'vk-change-user-play-status' );
10.    $ajax_nonce = wp_create_nonce( "vk-my-special-string" );
11.    wp_localize_script( 'vk-change-user-play-status', 'vk_select_params',
    array(
12.        'vk_ajax_url' => admin_url( 'admin-ajax.php' ),
13.        'ajax_nonce'  => wp_create_nonce( "vk-my-special-string" ),
14.        'plugin_url'  => plugins_url()
15.    ) );
16.    }
17.
18. add_action( 'wp_ajax_update_seen', 'update_seen_callback' );
19. function update_seen_callback() {
20.     global $wpdb;
21.     check_ajax_referer( 'vk-my-special-string', 'security' );
22.
23.     $user_id = intval( $_POST['user_id'] );
24.     $play_id = intval( $_POST['play_id'] );

```

```

25.     $status_id = intval( $_POST['status_id'] );
26.
27.     $result = $wpdb->get_row($wpdb->prepare("SELECT * FROM {$wpdb->
    >prefix}vk_user_play_status WHERE user_id = %d AND play_id = %d", $user_id,
    $play_id));
28.     $res_count = count($result);
29.
30.     if ( $res_count && $result->status_value == 0 ) {
31.         $wpdb->update(
32.             $wpdb->prefix . 'vk_user_play_status',
33.             array( 'status_value' => 1 ),
34.             array( 'status_id' => $result->status_id ),
35.             array( '%d' )
36.         );
37.         echo '1';
38.     } elseif( $res_count && $result->status_value == 1 ) {
39.         $wpdb->delete(
40.             $wpdb->prefix . 'vk_user_play_status',
41.             array(
42.                 'user_id' => $user_id,
43.                 'play_id' => $play_id
44.             ), array( '%d', '%d' )
45.         );
46.         echo '-1';
47.     } else {
48.         $wpdb->insert(
49.             $wpdb->prefix . 'vk_user_play_status',
50.             array(
51.                 'user_id' => $user_id,
52.                 'play_id' => $play_id,
53.                 'status_value' => 1
54.             ),
55.             array( '%d', '%d', '%d' )
56.         );
57.         echo '1';
58.     }
59.     wp_die();
60. }
61.
62. add_action( 'wp_ajax_want_to_see', 'want_to_see_callback' );
63. function want_to_see_callback() {
64.     global $wpdb;
65.     check_ajax_referer( 'vk-my-special-string', 'security' );
66.
67.     $user_id = intval( $_POST['user_id'] );
68.     $play_id = intval( $_POST['play_id'] );
69.     $status_id = intval( $_POST['status_id'] );
70.
71.     $result = $wpdb->get_results($wpdb->prepare("SELECT * FROM {$wpdb->
    >prefix}vk_user_play_status WHERE user_id = %d AND play_id = %d", $user_id,
    $play_id));
72.     $res_count = count($result);
73.
74.     if ( $res_count && $result->status_value == 1 ) {
75.         $wpdb->update(

```

```

76.         $wpdb->prefix . 'vk_user_play_status',
77.         array( 'status_value' => 0 ),
78.         array( 'status_id' => $status_id ),
79.         array( '%d' )
80.     );
81.         echo '0';
82.     } elseif ( $res_count && $result->status_value == 0 ) {
83.     $wpdb->delete(
84.         $wpdb->prefix . 'vk_user_play_status',
85.         array( 'user_id' => $user_id,
86.             'play_id' => $play_id), array( '%d', '%d' )
87.     );
88.         echo '-1';
89.     } else {
90.         $wpdb->insert(
91.             $wpdb->prefix . 'vk_user_play_status',
92.             array(
93.                 'user_id' => $user_id,
94.                 'play_id' => $play_id,
95.                 'status_value' => 0
96.             ),
97.             array( '%d', '%d', '%d' )
98.         );
99.         echo '0';
100.     }
101.     wp_die();
102. }

```

3.3. Изграждане на средства за навигация по актьори, събития, театри

3.3.1. Навигация по актьори

Изключително удобно за по-късно навигиране би било ако всеки актьор се записва като етикет на поста. Това обаче в реална ситуация с над 20 участващи артиста би било много трудоемко и изморително – дори и да оставим настрана въпроса, че зад една постановка няма само актьори, а и много други хора, грижещи се за музиката, костюмите и т.н. Но пък може да се съчетаят две от нещата, които вече ги има на разположение – атрибутите за продукт и етикетите.

След всяко натискане на бутона „Запис“ при атрибутите, се извикват функции, която ги обновява в базата данни. Същите тези функции може да използваме и разширим, за да запазим стойността на всеки атрибут като т.н.

категоризация (taxonomy) на продукта/пиесата, както всъщност се запазват в базата етикетите. По този начин ще имаме възможността свободно да запазим редакцията на екипа и едновременно с това, ще можем автоматично да генерираме страници за всеки актьор и член на екипа зад постановката.

Самото записване правим като вземем всички стойности на атрибутите, обходим масива, в който се държат и за всяка стойност, ако има повече от едно записано име, обхождаме нов масив, като за всяко име създаваме връзка между term и taxonomy, посредством wp_set_object_terms() функцията на WordPress:

```
1. $tags = isset( $data['attribute_values'] ) ? $data['attribute_values'] :  
    array();  
2.  
3. foreach ( $tags as $tag ) {  
4.     $names = explode( ',', $tag );  
5.  
6.     foreach ( $names as $value ) {  
7.         wp_set_object_terms( $post_id, $value, 'product_tag', true );  
8.     }  
9. }
```

Правим извеждането на стойностите. При обикновеното показване от WooCommerce, стойностите само се изреждат със запетайки. Необходимо е освен това, да имаме и хиперлинк, водещ до страницата на члена на екипа.

В шаблона на продукта/пиесата вместо просто да показваме стойностите:

```
1. echo apply_filters( 'woocommerce_attribute', wpautop( wptexturize( implode(  
    ', ', $values ) ) ), $attribut, $values );
```

е необходимо да се вземе и линка, водещ до тази класификация и да се обработи така, че да се изведе правилно:

```
1. $values = array_map(  
2.     'trim', explode( ',', $attribute['value'] ) );  
3.  
4. foreach ( $values as $key => &$value ) {  
5.     $value = '<a href="' . get_term_link( $value, 'product_tag' ) . '">  
    . $value . '</a>';  
6. }  
7.  
8. echo apply_filters(  
9.     'vk_attribute',
```

```
10.     wpautop( wptexturize( implode( ' ', $values ) ) ),
11.     $values );
```

Показването на етикетите като такива на страницата на постановката вече става излишно, затова се махат.

Списък с всички актьори и екип зад пиесите може да се намери на страницата „Актьори“ от главното меню на сайта.

Самият списък се генерира като ползваме функцията на WordPress `wp_list_categories()` и избираме всички класификацията „product_tag“.

```
1. <?php
2. $args = array(
3.     'show_option_all'    => '',
4.     'orderby'            => 'name',
5.     'order'              => 'ASC',
6.     'style'              => 'list',
7.     'show_count'         => 0,
8.     'hide_empty'         => 1,
9.     'use_desc_for_title' => 1,
10.    'child_of'            => 0,
11.    'feed'                => '',
12.    'feed_type'           => '',
13.    'feed_image'          => '',
14.    'exclude'             => '',
15.    'exclude_tree'        => '',
16.    'include'             => '',
17.    'hierarchical'        => 1,
18.    'title_li'            => __( '' ),
19.    'show_option_none'    => __( '' ),
20.    'number'              => null,
21.    'echo'                => 1,
22.    'depth'               => 0,
23.    'current_category'    => 0,
24.    'pad_counts'          => 0,
25.    'taxonomy'            => 'product_tag',
26.    'walker'              => null
27. );
28.
29. wp_list_categories( $args );
30. ?>
```

3.3.2. Навигация по театри

Навигацията по театри, а и отбелязването на дадена пиеса като изпълнявана в даден театър се случва посредством категориите на поста. Като

родителска категория се отбелязва града, в който се намира театъра, а като под категория – театъра.

На страницата на всяка пиеса може да се видят категориите, в които присъства. За да изглеждат по-добре, надписът „Категории“ се маха, а на негово място е поставена иконка от Glyphicons, указваща местоположение:

```
1. echo '<span class="glyphicon glyphicon-map-marker"></span>';  
2. echo $product->get_categories(  
3.     ' ', ' ', '<span class="posted_in">' . ' ', '</span></br>'  
    );
```

При кликване на града се отваря страница с всички театри, намиращи се в него, за които в базата има записани пиеси.

Освен това списък на всички градове и театри в тях може да се намери на страницата „Театри“. За целта в директорията на темата е създаден файл с име `page-theatres.php`. В него чрез функцията на WordPress `wp_list_categories()` като линкове извеждаме всички категории по почти същия начин като актьорите, с разликата, че този път избраната таксономия е „`product_cat`“.

```
1. <?php the_post_thumbnail( 'activello-featured', array( 'class' => 'single-  
   featured' )); ?>  
2.  
3. <div class="post-inner-content">  
4. <article id="post-<?php the_ID(); ?>" <?php post_class(); ?>  
5.     <header class="entry-header page-header">  
6.         <h1 class="entry-title"><?php the_title(); ?></h1>  
7.     </header><!-- .entry-header -->  
8.  
9. </article><!-- #post-## -->  
10.</div>  
11.  
12.<?php  
13.     $args = array(  
14.         'show_option_all'      => '',  
15.         'orderby'              => 'name',  
16.         'order'                => 'ASC',  
17.         'style'                => 'list',  
18.         'show_count'           => 0,  
19.         'hide_empty'           => 1,  
20.         'use_desc_for_title'   => 1,  
21.         'child_of'             => 0,  
22.         'feed'                 => '',  
23.         'feed_type'            => '',  
24.         'feed_image'           => '',
```

```

25.     'exclude'           => '',
26.     'exclude_tree'      => '',
27.     'include'           => '',
28.     'hierarchical'      => 1,
29.     'title_li'          => __( '' ),
30.     'show_option_none'  => __( '' ),
31.     'number'            => null,
32.     'echo'              => 1,
33.     'depth'             => 0,
34.     'current_category'  => 0,
35.     'pad_counts'        => 0,
36.     'taxonomy'          => 'product_cat',
37.     'walker'            => null
38. );
39.
40. wp_list_categories( $args );
41. ?>

```

3.3.3. Навигация по събития/пиеси

Навигацията по събития се случва по няколко начина: чрез програмата на началната страница, чрез програмата в страничната лента и чрез страниците на архива.

3.3.3.1. Начална страница

Началото на сайта също е специално написана страница в директорията на темата. Това е файлът `page-index.php`. В него първо вземаме текущото местоположение на потребителя и според него и днешната дата от базата извеждаме всички пиеси, които предстоят да се изиграят тогава.

Гео-локацията вземаме с разширението `geoPlugin`^[24]:

```

1. $geoplugin = new geoPlugin();
2. $geoplugin->locate();
3. $city = $geoplugin->city;

```

След което тя се обработва заедно с днешната дата и се извежда на страницата:

```

1. <ol class="breadcrumb">
2.   <li><a href="/">Начало</a></li>
3.   <li><a href="#">Театри в <?php
4.     echo vk_latin_cyrillic( $city );
5.   ?></a></li>

```



```

6.     <li class="active">
7.         <?php
8.             $value = (string)date("d/m/Y");
9.             vk_process_date( $value );
10.        ?>
11.    </li>
12.</ol>

```

Изпраща се заявка до базата данни и се вземат всички пиеси/събития, които отговарят на условията да са във взетото местоположение (т.е. – имат категория град) и да имат представление на днешния ден.

Извеждат се на страницата в подменюто „Днес“ по 3 на ред, с изображение, заглавие, театър и начален час. Абсолютно същото се случва в „Утре“, но разбира се, с утрешна дата.

```

1. <?php
2. $count = 1;
3. if ($pageposts): ?>
4.     <?php global $post;?>
5.     <?php foreach ($pageposts as $post): ?>
6.         <?php
7.             $html = woocommerce_get_product_thumbnail( 'shop_catalog' );
8.             $xpath = new DOMXPath(@DOMDocument::loadHTML($html));
9.             $img_src = $xpath->evaluate("string(//img/@src)");
10.        ?>
11.        <?php if ($count == 1) {
12.            echo '<div class="row">';
13.        }
14.        ?>
15.        <div class="col-xs-6 col-md-4">
16.        <a href="<?php the_permalink() ?>" >
17.            ">
19.            </a>
20.            <h4 class="title"><a href="<?php the_permalink() ?>" ><?php the_title();
21.            ?></a></h4>
22.            <small class="text-muted">
23.                <?php
24.                    $meta_time = get_post_meta( $post->ID,
25.                    '_vk_time_input', true );
26.                    if($meta_time)
27.                        echo '<span class="glyphicon glyphicon-
28.                        time"></span> ' . $meta_time . '<br/>';
29.                    $taxonomyName = "product_cat";
30.                    $parent_terms = wp_get_post_terms( $post->ID,
31.                    $taxonomyName, array('parent' => 0, 'orderby' => 'slug', 'hide_empty' =>
32.                    false));

```

```

29.             echo '<span class="glyphicon glyphicon-map-
marker"></span>';
30.
31.         foreach ($parent_terms as $pterm) {
32.             $terms = wp_get_post_terms( $post->ID, $taxonomyName,
array('parent' => $pterm->term_id, 'orderby' => 'slug', 'hide_empty' =>
false));
33.             foreach ($terms as $term) {
34.                 echo '<a href="' . get_term_link( $term->name,
$taxonomyName ) . '">' . $term->name . '</a>';
35.             }
36.         }
37.         ?>
38.         </small>
39.         <hr/>
40.     </div>
41.     <?php if ($count == 3) {
42.         echo '</div>';
43.         $count = 0;}
44.     $count++; ?>
45. <?php endforeach;
46. if(count($pageposts) % 2 == 0 || count($pageposts) == 1) {
47.     echo "</div>";
48. }
49. ?>
50. <?php else : ?>
51.     <h2 class="center">Няма резултати</h2>
52.     <p class="center">Съжалявам, търсите нещо, което не е тук.</p>
53.     <?php include (TEMPLATEPATH . "/searchform.php"); ?>
54. <?php endif;

```

3.3.3.2. Навигация от програмата

За изграждане на програмата се използва jQuery UI и по-точно – DatePicker^[25]. Джаджата позволява много лесно да се направи обикновен календар, с който да може да се избира дата.

След добавянето на скрипта към страницата, създаването е лесно:

```

1. <div id="calendar"></div>

```

С JavaScript се подават подходящи аргументи и отговора от DatePicker се обработва, за да се превърне в хиперлинк:

```

1. (function($) {
2.     $('#calendar').datepicker({
3.         inline: true,
4.         firstDay: 1,
5.         showOtherMonths: false,

```

```

6.     dayNamesMin: ['Нд', 'Пн', 'Вт', 'Ср', 'Чт', 'Пт', 'Сб'],
7.     dateFormat: "dd/mm/yy",
8.     onSelect: function(date) {
9.         var getUrl = window.location;
10.        var baseUrl = getUrl .protocol + "//" + getUrl.host + "/" +
        getUrl.pathname.split('/')[1];
11.        window.location.href = baseUrl + "/bydate?&date=" + date;
12.    }
13. });
14. })(jQuery);

```

Датата отвежда към страница, почти същата като началната, с разликата, че има добавено под-меню за избраната дата, където се поместват резултатите.

3.4. Търсене върху актьори, събития, театри

3.4.1. Търсене

По подразбиране търсенето в WordPress е насочено към намиране в заглавието или съдържанието на поста. Това макар и да помага в директното търсене на постановки, пречи при актьорите и театрите, понеже те не се записват в заглавието или съдържанието. Затова се налага да „разширим“ обикновеното търсене, за да включва и персонализирани полета (custom fields).

Цялата информация относно персонализираните полета се държи в таблицата „postmeta“ в базата. По подразбиране търсенето на WordPress е насочено само към таблица „posts“. За да включим и необходимите полета трябва да направим ляво съединяване (LEFT JOIN) на таблиците „posts“ и „postmeta“ в базата данни.

```

1. function vk_search_join( $join ) {
2.     global $wpdb;
3.
4.     if ( is_search() ) {
5.         $join .= ' LEFT JOIN ' . $wpdb->postmeta . ' ON ' . $wpdb->posts . '.ID
= ' . $wpdb->postmeta . '.post_id ';
6.     }
7.
8.     return $join;
9. }
10. add_filter('posts_join', 'vk_search_join' );

```

След това трябва да се промени заявката, за да може да се добавят и персонализираните полета:

```
1. function vk_search_where( $where ) {
2.     global $wpdb;
3.
4.     if ( is_search() ) {
5.         $where = preg_replace(
6.             "/\(\s*".$wpdb->posts.">post_title\s+LIKE\s*(\'[^\']*+\')\s*\)/",
7.             "(\".$wpdb->posts.">post_title LIKE $1) OR (\".$wpdb->postmeta.">meta_value LIKE $1)", $where );
8.     }
9.
10.    return $where;
11. }
12. add_filter( 'posts_where', 'vk_search_where' );
```

За финал към SQL заявката трябва да се добави ключовата дума „DISTINCT“, за да се предотврати връщането на дубликати:

```
1. function vk_search_distinct( $where ) {
2.     global $wpdb;
3.
4.     if ( is_search() ) {
5.         return "DISTINCT";
6.     }
7.
8.     return $where;
9. }
10. add_filter( 'posts_distinct', 'vk_search_distinct' );
```

3.5. Бюлетин и изпращане на електрона покана до приятел

3.5.1. Бюлетин

Бюлетинът е функционалност, която съществува в почти всички сайтове в Интернет днес. Благодарение на него потребителите могат да научават новостите директно на електронната си поща.

В настоящата дипломна работа тази функция се реализира с външната услуга MailChimp. По този начин администраторът може да избере точно каква

информация и по какъв начин поднесена да се изпрати на абонираните се потребители.

Самият абонамент от платформата се извършва през форма в страничната лента:

```
1. <aside id="digest" class="widget">
2.     <h3 class="widget-title">Абонирайте се за нашия бюлетин</h3>
3.
4.     <link href="//cdn-images.mailchimp.com/embedcode/slim-10_7.css"
5.     rel="stylesheet" type="text/css">
6.     <style type="text/css">         #mc_embed_signup{background:#fff;
7.     clear:left; font:14px Helvetica,Arial,sans-serif; }</style>
8.     <div id="mc_embed_signup">
9.         <form action="//byethost13.us12.list-
10.         manage.com/subscribe/post?u=5553e50301d77b7c5d47aa287&id=257f5d9ce8"
11.         method="post" id="mc-embedded-subscribe-form" name="mc-embedded-subscribe-
12.         form" class="validate" target="_blank" novalidate>
13.             <div id="mc_embed_signup_scroll">
14.                 <input type="email" value=""
15.                 name="EMAIL" class="email" id="mce-EMAIL" placeholder="мейл адрес"
16.                 required>
17.                 <div style="position: absolute; left: -
18.                 5000px;" aria-hidden="true"><input type="text"
19.                 name="b_5553e50301d77b7c5d47aa287_257f5d9ce8" tabindex="-1" value=""></div>
20.                 <div class="clear"><input type="submit"
21.                 value="Запис" name="subscribe" id="mc-embedded-subscribe"
22.                 class="button"></div>
23.             </div>
24.         </form>
25.     </div>
26. </aside>
```

3.5.2. Изпращане на електронна покана до приятел

Формата за изпращане на такава покана се намира в страничната лента до всяка постановка.

Формата се състои от поле за подател, емайл на получател и кратко съобщение. Самите емайли се изпращат през SMTP услуга, ползвайки PHP Mailer.^[26]

```
1. <aside id="invitation" class="widget">
2.
3.     <h3 class="widget-title">Покани приятел</h3>
4.
5.     <form action="" method="post">
6.         От:<br/><input type="text" name="name"><br>
7.         До:<br/><input type="text" name="email"><br>
```

```

8.      Съобщение:<br/><textarea rows="5" name="message"
      cols="30"></textarea><br/><br/>
9.
10.      <input type="submit" name="submit" value="Изпрати">
11.  </form>
12.
13.  <?php
14.
15.      if(isset($_POST['submit'])){
16.
17.          $to = $_POST['email'];
18.          $name = $_POST['name'];
19.          $subject = "Покана за театър";
20.          $message = $_POST['message'];
21.
22.          $subject = $name . ' Ви покани да гледате "' . get_the_title(
      $post->ID ) . '" заедно' ;
23.
24.          $content = '<h4>' . $name . ' Ви покани да гледате "' .
      get_the_title( $post->ID ) . '" заедно</h4></bg>
25.          <p>' . $message . '</p>' . get_post_field("post_content",
      $post->ID);
26.
27.          date_default_timezone_set('Etc/UTC');
28.
29.          require 'PHPMailer/PHPMailerAutoload.php';
30.
31.          $mail = new PHPMailer;
32.
33.          $mail->isSMTP();
34.          $mail->Host = 'smtp.gmail.com';
35.          $mail->SMTPAuth = true;
36.          $mail->Username = '<email>@gmail.com';
37.          $mail->Password = <password>;
38.
39.          $mail->SMTPSecure = 'tls';
40.          $mail->Port = 587;
41.          $mail->setFrom('<email>@gmail.com', 'bgscena');
42.          $mail->addAddress( $to );
43.
44.          $mail->isHTML(true);
45.
46.          $mail->Subject = $subject;
47.          $mail->Body      = $post->$content . '</br>' . $content;
48.
49.          if(!$mail->send()) {
50.              echo 'Съобщението не можа да бъде изпратено.';
51.              echo 'Грешка: ' . $mail->ErrorInfo;
52.          } else {
53.              echo 'Съобщението бе изпратено.';
54.          }
55.      }
56.  }
57.  ?>
58.  </aside>

```

На много места в процеса на разработка в имената на функции и даже таблица в базата данни е употребена представката „vk_”. Това е направено с цел сигурност и защита на кода от конфликти с функции на WordPress, други разширения и теми и се препоръчва от разработчиците на CMS-a, а самата представка е инициалите на дипломанта.

ГЛАВА IV

РЪКОВОДСТВО НА ПОТРЕБИТЕЛЯ

4.1. Въвеждане на информация

Всеки потребител с необходимото ниво правомощия може да въвежда информация за нови пиеси. Това най-лесно става от административния панел или директно от административната лента:

1. „Постановки“ - > „Добави постановка“
2. „Създаване“ - > „Постановка“

В полето за име се записва заглавието на пиесата. Директно под него се намира полето за резюме/описание.

Следва допълнителната информация. Натиснете бутона „Добавяне“, за да заредите колкото полета са необходими за изреждане на екипа зад пиесата, като в „име“ запишете наименованието на ролята, а в „стойност“ – името. Ако има повече от един човек за дадена роля, имената се разделят със запетайки (фиг. 4.1.1.).

Име:	Стойност(и):
<input type="text" value="Участват"/>	Лилия Маравиля, Дария Симеонова, Лора Мутишева, Невена Калудова, Силвия Петкова, Асен Мутафчиев, Юлиан Рачков, Николай Върбанов, Симона Халачева, Антон Григоров, Юлиан Малинов, Майя Остоич, Йорданка Любенова, Николай Димитров, Иринеи Константинов, Росен Белов

Фигура 4.1.1. Въвеждане на атрибут

След въвеждане на цялата информация се натиска бутона „Запис“.

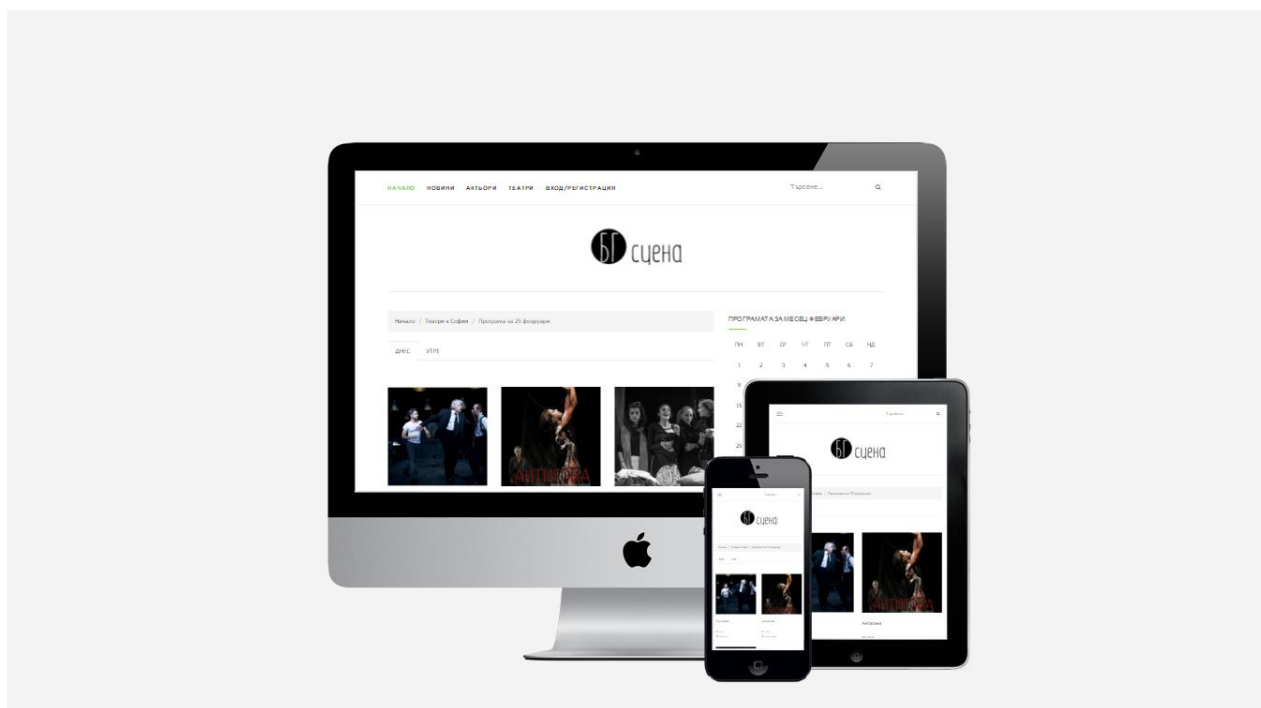
В кутийката за детайли се записват датите, времето, сцената и линк към сайт, от където могат да се закупят билети. Датите се записват във формат ДД/ММ/ГГГГ и се разделят със запетайки.

В кутията „Категории“ се отбелязва града, в който се намира театъра и под него самия театър. Ако не съществуват такива категории може да се създадат.

Избира се изображение на постановката, което има ролята на афиш. Под него е полето за галерия на постановката, където могат да се качат няколко изображения.

Когато всичко е готово натиснете бутона „Публикуване“.

4.2. Начална страница



Фигура 4.2.1. Визуализация на началната страница на платформата на различни устройства

4.2.1. Главна част

В главната част на началната страница могат да се видят всички постановки, които ще се изиграят на днешния ден, а при натискане на другия таб – тези за следващия ден.

Като информация за всяка пиеса е показан умален афиш/корица, заглавието на пиесата, от колко часа ще започне и къде.

4.2.2. Странична лента

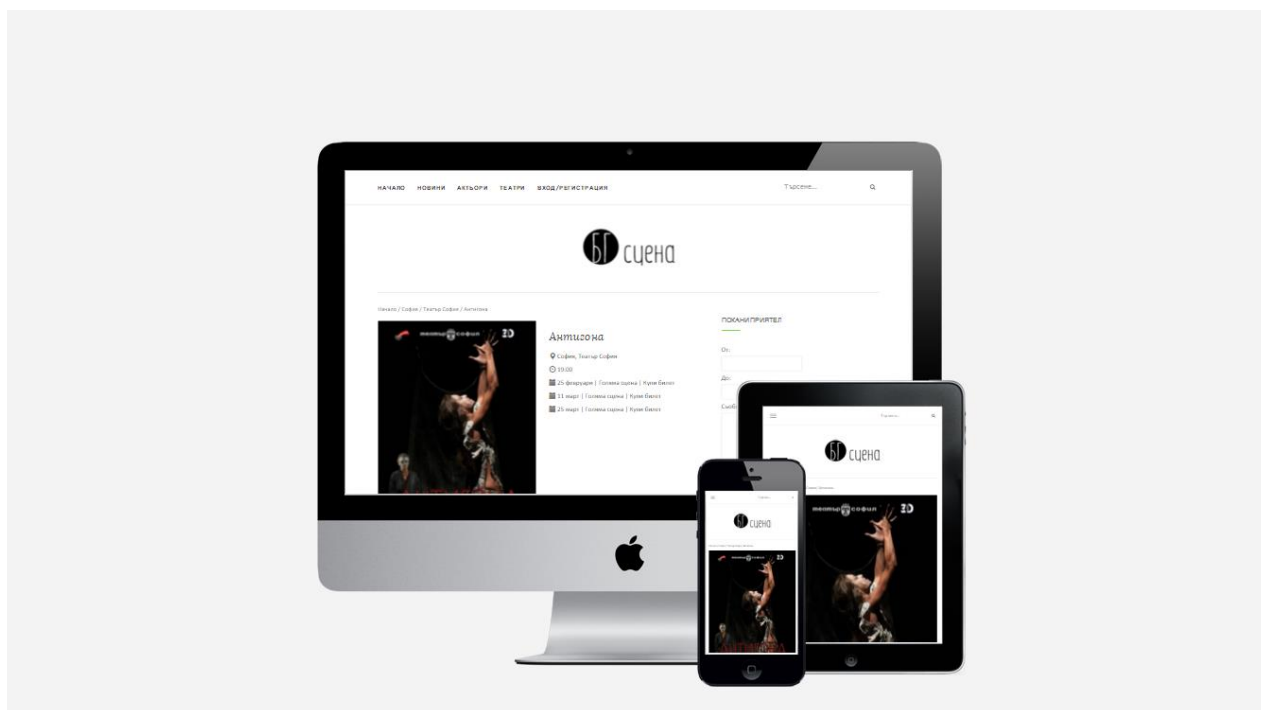
В страничната лента се намира календар, който е програма за месеца. При натискане на някоя от датите се зарежда страница, на която могат да се видят постановките за този ден, подобно на визуализирането на тези в главната част.

Под програмата за месеца се намира малко поле, където се визуализират последните коментари написани в сайта, както и за коя пиеса се отнася всеки от тях.

Следва малка форма за записване към емайл бюлетин.

Последна е малката джаджа с най-новите страници архиви по месеци.

4.3. Страница на постановката



Фигура 4.3.1. Визуализация на страница на пиеса на различни устройства, видяна от нерегистриран потребител

4.3.1. Постановката

Страницата на всяка постановка се състои от няколко главни елемента:

- Заглавие;
- Обложка;
- Галерия;
- Средна оценка и брой гледали;
- Място, дата и време на изпълнение;
- Полета за описание, допълнителна информация и ревьюта;
- Подобни постановки.

4.3.2. Бутони за статус

Всеки регистриран потребител може да отбележи всяка постановка като „Гледана“ или „Искам да гледам“. Това се случва от страницата на

постановката чрез натискане на зелените бутони. Ако повторно се натисне бутон, с който вече е отбелязан даден статус, то записът ще се изтрие.

4.3.3. Странична лента

4.3.3.1. Покани приятел

Джаджата Ви позволява да изпратите емайл покана на Ваш приятел за пиесата. Необходимо е да въведете своето име, емайл адреса на приятел и ако желаете да добавите съобщение. След това натиснете „Изпрати“.

4.3.3.2. Приятели, които...

Тази джаджа се появява само ако има такива. Ако имате приятели в сайта, които са гледали пиесата, техните профилни снимки ще се покажат под надпис „Приятели, гледали пиесата“. Ако пък имате приятели, които имат желание да я гледат, те ще са под „Приятели, които искат да гледат пиесата“. В случай, че имате и гледали и желаещи, то и двете полета ще са там.

4.4. Потребителски профили

От падащото меню „Моят профил“ може да се навигира през страниците Профил, Моите пиеси, Приятели, Споменавания, Съобщения и Изход. От „Моите пиеси“ можете да разгледате всички пиеси, които сте отбелязали заедно с малко информация за тях.

В профилите се съхраняват Вашите споменавания, приятели и скорошни активности.

Можете да видите пиесите на потребителя.

4.5. Страниците „Актьори“, „Театри“ и „Архив“

Не е необходимо да сте регистриран потребител, за да имате достъп до тези страници.

4.5.1. Страницата „Актьори“

На тази страница можете да видите всички актьори, участвали в пиесите, записани в базата данни на платформата. Листът е подреден по азбучен ред.

Всяко име води към персонализирана страница, в която могат да се намерят всички участия на дадения артист.

4.5.2. Страницата „Театри“

Страницата показва всички театри, за които има информация в базата данни подредени по азбучен ред по градове.

Всеки елемент от листа води към страница. Ако елементът е град – страницата съдържа всички театри в този град. Ако е театър – страницата, която се зарежда съдържа всички пиеси, за които има информация, че се играят там.

4.5.3. Страниците „Архив“

В тези страници по месеци могат да се видят всички новини и публикувани постановки.

4.6. Търсене

От полето за търсене в горната част на платформата могат да се търсят както актьори, така и пиеси. Ако търсенето е за актьор, като резултат ще се бъдат изведени всички постановки, в които има някакво участие.

ЗАКЛЮЧЕНИЕ

В настоящата дипломна работа бяха разгледани множество страни от създаването на онлайн платформи.

Беше разработена онлайн платформа от тип социален каталог с насоченост български театри. Нуждата от това нараства поради навлезлите на мода нови технологии и все по-малко време за губене в търсене на информация.

Софтуерният продукт е с богата функционалност, но има модули, които могат да бъдат усъвършенствани и други, които да бъдат добавени. Като бъдещо развитие на дипломната работа може да се включат интеграция на платформата с лични календари от типа на Google Calendar, Outlook Calendar и др., както и изграждане на вътрешна система за продажба на билети.

ИЗПОЛЗВАНА ЛИТЕРАТУРА

1. Пенева, Е. (9 4 2009 г.). Защо да ходим на театър. Извлечено от <http://www.dnes.bg/health/2009/04/09/zashto-da-hodim-na-teatyr.69074>
2. Chandler, О. (23 07 2013 г.). Goodreads Grows to 20 Million Readers. Извлечено от <https://www.goodreads.com/blog/show/425-goodreads-grows-to-20-million-readers>
3. (н.д.). Извлечено от IMDb - Movies, TV and Celebrities - IMDb: <http://www.imdb.com/>
4. (н.д.). Извлечено от Last.fm: <http://www.last.fm/>
5. (н.д.). Извлечено от theatre.art.bg: <http://theatre.art.bg/>
6. (н.д.). Извлечено от програмата: <http://www.programata.bg/>
7. (н.д.). Извлечено от след5: <http://www.dnevnik.bg/sled5/>
8. Usage Statistics and Market Share of Server-side Programming Languages for Websites, February 2016. (02 2016 г.). Извлечено от http://w3techs.com/technologies/overview/programming_language/all
9. Какво е CMS. (н.д.). Извлечено от <https://host.bg/bg/info/CMS>
- 10.(н.д.). Извлечено от PHP: Hypertext Preprocessor: <http://php.net/>
- 11.CMS Usage Statistics. (02 2016 г.). Извлечено от <http://trends.builtwith.com/cms>
- 12.Spencer, J. (06 18 2015 г.). CONTENT MANAGEMENT SYSTEM COMPARISON – WORDPRESS VS JOOMLA VS DRUPAL. Извлечено от <http://makeawebsitehub.com/content-management-system-cms-comparison/>
- 13.Colao, J. (2 9 2012 г.). With 60 Million Websites, WordPress Rules The Web. So Where's The Money? Извлечено от Forbs: <http://www.forbes.com/sites/jjcolao/2012/09/05/the-internets-mother-tongue/>
- 14.Joomla срещу Drupal. (25 08 2014 г.). Извлечено от <http://mdesign-bg.com/%D0%BF%D0%BE%D0%BB%D0%B5%D0%B7%D0%BD%D0%BE/252-joomla-%D1%81%D1%80%D0%B5%D1%89%D1%83-drupal>

15. Activello – Simple Multipurpose Blog Theme. (н.д.). Извлечено от Colorlib:
<https://colorlib.com/wp/themes/activello/#1449773513719-4b122e54-f4bf>
16. WooCommerce - a free eCommerce toolkit for WordPress. (н.д.). Извлечено от WooThemes: <https://www.woothemes.com/woocommerce/>
17. WooCommerce. (н.д.). Извлечено от
<https://wordpress.org/plugins/woocommerce/>
18. BuddyPress. (н.д.). Извлечено от WordPress:
<https://wordpress.org/plugins/buddypress/>
19. JavaScript Usage Statistics. (02 2016 г.). Извлечено от Built With:
<http://trends.builtwith.com/javascript>
20. (н.д.). Извлечено от jQuery: <https://jquery.com/>
21. Search · stars:>1. (н.д.). Извлечено от GitHub:
<https://github.com/search?l=&o=desc&q=stars%3A%3E1&ref=advsearch&s=stars&type=Repositories>
22. (2016). Извлечено от WordPress Codex: <http://codex.wordpress.org/>
23. (н.д.). Извлечено от WordPress Developer Resources:
<https://developer.wordpress.org/>
24. (н.д.). Извлечено от geoPlugin: <http://www.geoplugin.com/>
25. Datepicker | jQuery UI. (н.д.). Извлечено от jQuery UI:
<https://jqueryui.com/datepicker/>
26. PHPMailer/PHPMailer. (н.д.). Извлечено от
<https://github.com/PHPMailer/PHPMailer>

СЪДЪРЖАНИЕ

УВОД	4
ГЛАВА I	6
ПРОУЧВАНЕ НА ПОДОБНИ ОНЛАЙН ПЛАТФОРМИ. ТЕХНОЛОГИИ ЗА ТЯХНАТА РЕАЛИЗАЦИЯ	6
1.1. Социалните каталози в световен мащаб	6
1.1.1. Goodreads	6
1.1.2. Internet Movie Database	7
1.1.3. Last.fm	7
1.2. Български платформи насочени към театралната сцена	8
1.2.1. Theatre.art.bg	8
1.2.2. Teatri.bg	9
1.2.3. Програмата.....	10
1.2.4. След5	11
1.3. Развойни средства и среди	12
ГЛАВА II	15
ИЗИСКВАНИЯ КЪМ ПРОГРАМНИЯ ПРОДУКТ. ИЗБОР НА РАЗВОЙНА СРЕДА И СРЕДСТВА	15
2.1. Изисквания	15
2.2. Избор на програмни езици	15
2.2.1. PHP	15
2.2.2. JavaScript	16
2.3. Сравнение между WordPress, Joomla и Drupal	17
2.3.1. WordPress	18
2.3.2. Drupal.....	19
2.3.3. Joomla	20

2.4.	Допълнителни средства	21
2.4.1.	Activello.....	21
2.4.2.	WooCommerce	22
2.4.3.	BuddyPress.....	23
2.4.4.	jQuery.....	24
2.4.5.	jQuery UI	25
2.4.6.	Bootstrap	25
2.5.	Структура на базата данни	25
2.5.1.	MySQL.....	25
2.5.2.	Структура на базата	26
ГЛАВА III.....		29
ПРОГРАМНА РЕАЛИЗАЦИЯ НА ОНЛАЙН ПЛАТФОРМА, СОЦИАЛЕН		
КАТАЛОГ С НАСОЧЕНОСТ БЪЛГАРСКИ ТЕАТРИ.....		29
3.1.	Управление на процеса по въвеждане на данни	29
3.2.	Включване към събития	31
3.3.	Изграждане на средства за навигация по актьори, събития, театри	36
3.3.1.	Навигация по актьори.....	36
3.3.2.	Навигация по театри	38
3.3.3.	Навигация по събития/пиеси	40
3.3.3.1.	Начална страница	40
3.3.3.2.	Навигация от програмата.....	42
3.4.	Търсене върху актьори, събития, театри.....	43
3.4.1.	Търсене.....	43
3.5.	Бюлетин и изпращане на електрона покана до приятел.....	44
3.5.1.	Бюлетин.....	44
3.5.2.	Изпращане на електронна покана до приятел.....	45

ГЛАВА IV	48
РЪКОВОДСТВО НА ПОТРЕБИТЕЛЯ	48
4.1. Въвеждане на информация	48
4.2. Начална страница	49
4.2.1. Главна част	50
4.2.2. Странична лента	50
4.3. Страница на постановката	50
4.3.1. Постановката	51
4.3.2. Бутони за статус	51
4.3.3. Странична лента	52
4.3.3.1. Покани приятел	52
4.3.3.2. Приятели, които	52
4.4. Потребителски профили	52
4.5. Страниците „Актьори“, „Театри“ и „Архив“	53
4.5.1. Страницата „Актьори“	53
4.5.2. Страницата „Театри“	53
4.5.3. Страниците „Архив“	53
4.6. Търсене	53
ЗАКЛЮЧЕНИЕ	54
ИЗПОЛЗВАНА ЛИТЕРАТУРА	55
СЪДЪРЖАНИЕ	57
ПРИЛОЖЕНИЕ	60

ПРИЛОЖЕНИЕ