

**ACADEMIA DE STUDII ECONOMICE - București**  
**Bucharest University of Economic Studies**

**FACULTY OF BUSINESS ADMINISTRATION**  
(Facultatea de Administrare a Afacerilor cu predare în limbi străine)

# **Technologies for eBusiness**

-

## **CONTENT MANAGEMENT SYSTEMS**

**By: Professor Vasile AVRAM, PhD**

- suport de curs destinat studenților de la secția engleză -  
(course notes for 1<sup>st</sup> year students of English division)

- anul I - Zi -

**București 2013**



# Content Management Systems

## Contents

- 6 Content Management Systems..... 4
  - 6.1 Introduction ..... 4
  - 6.2 CMS Application ..... 4
  - 6.3 Open Source CMS Architecture and Functionality ..... 7
  - 6.4 Setup and Installing Locally Open Source CMS Solutions ..... 8
    - Setup WAMP stack..... 8
    - Setup WordPress Module ..... 13
    - Setup Joomla Module ..... 17
- Annex 1. Sample CMS ROI Calculator [Gs-06]..... 18
- References ..... 19

## 6 Content Management Systems

### 6.1 Introduction

When **www** created the only technology used to create the web pages represented by HTML. To do that the page developers make use traditional editors such as *Emacs* or *vi* on Unix/ Linux machines or *Microsoft Notepad* on Windows machines. These requires from the part of users the knowledge of HTML tags together with their attributes.

Later on, a new generation of web pages specialized editors, provided with graphical user interface working as “What You See Is What You Get” (WYSIWYG), appears and offer to the developer assistance during page creation, such as Adobe Dreamweaver and Microsoft Expression Web. The appearance of other web technologies facilitating complex applications development such as Cascading Style Sheet, Scripting Languages, and XML allows defining web documents templates and the creation of web pages with a predefined layout. Despite all these advances in technology when the website grows in the number of manipulated/ managed resources becomes harder to maintain the menus, linking coherence, sitemap, and even to update or align pages to new standard. The page content creators still enforced to have minimal markup language knowledge and to learn how to use the specialized editors.

To solve all kind of problems appearing for such types of web content development and to ensure the access of domain workers to web content creation the large media publishers turned to a special type of software called **content management system (CMS)**.

### 6.2 CMS Application

In IT terms content management refers to an informatics system (together with its processes) allowing information creation, manipulation, management, publishing, and archiving. The created information typically passes through this lifecycle for a finite period of time. The system provides the necessary infrastructure for multiple people to effectively contribute content and collaborate throughout these lifecycles. CMS allows a collaborative manipulation of documents and other content and make web page content simple and friendly for non-it users.

The CMS application:

- Automate site content management;
- Separation of content into blocks;
- Separation of layout and design from content;
- Allows users (such non-IT writers and journalists) to contribute articles directly into the system via customer user interface accessible within a browser window so that the Web development become decentralized;
- Allows readers, as a non-homogenous class of users regarding IT knowledge (on a scale from “dummy” to “expert”), to contribute comments to articles which in turn are administered/ managed by non-IT users;
- The interface offered to the regular users do not requires at all the knowledge of HTML or having some specific IT skill. For sure the knowledge of HTML and CSS, even at a primary level, allows users to create more appealing articles by switching from the text editor or the WYSIWYG editor to the HTML one, when typing the article in the form given by the system;
- Reduces the amount of custom development and automates the workflow management;
- Makes the pages dynamic, it means a page doesn't really exist until requested (by following a link to view it, for example) and the content might be different each time viewed since page's content can be updated and customized based on the viewer's interactions with the page. The content administration is database driven.

The changes in web technologies and the necessity to adapt the website to these to gain/ do not losing the technological advantage requires changes of large amounts of web pages. By adopting CMS the only required changes are for upgrading the CMS application to the one incorporating those new technologies and without any changes to the content which will be formatted to the updated layouts in the same way for the oldest ones.

The CMS application included administrative features are compelling:

- Complete management possible via a robust web interface (allows a form-based content administration);
- Web-based management of site assets such as graphics, files, and other media;
- Content approval features allow moderating of remote author postings;
- User management and hierarchical user group management;
- Automated menu management;
- Content publication scheduling for automatic publishing and removal of articles;
- Integration with other servers including FTP, e-mail, and LDAP;
- Contributor posting management.

The CMS application content capabilities include the following:

- Multiple built-in “What You See Is What You Get” (WYSIWYG) editors;
- Automatic full text search of site content;
- Optional search engine–friendly (SEF) URLs to article content;
- Full support for newsfeeds in RSS or Atom format;
- Built-in user polling;
- Banner advertising management;
- Plug-ins for e-commerce solutions, including shopping cart, picture gallery, inventory management, and point of sale;
- Multilingual internationalization features;
- Accessibility options for the people with impairments.

All CMS open source solutions offers a good number of system advantages, including the following:

- Full open source license with free download of the application and source code;
- Availability on all major operating systems (Unix/ Linux, Mac OS, and Windows);
- Page caching for improved performance.

CMS provides many advantages over traditional methods of information editing. Many of the benefits are derived when distributed teams of people are responsible for coordinating and contributing to different content repositories. The main advantages taken includes the following [GS-06][HH-10]:

**1. Empower Content** - making better use and control of information:

- Content Ownership - empowers business users to quickly and efficiently update their online information without technical intervention (distribute content management to users);
- Content Accessibility (see <http://www.w3.org/WAI/WCAG1AAA-Conformance>) - repurposes content into multiple formats, and helps ensure disability compliance is met.

**2. Decrease Costs** - lowers costs associated with managing information online:

- Content Creation - less costly as business users can directly contribute information online without going through an intermediary;
- Content Management - less costly as content is maintained by business users and standard processes are automated;
- Content Publishing - less costly as information is scheduled in advance to be published at a specific date and time.

**3. Increase Revenues** - provides new ways to increase revenues:

- Time Sensitive Opportunities New opportunities arise when information is published online in a matter of minutes as compared to hours or days;
- Fresh Content Encourages Return Customers.

4. **Improve Accountability** - for both the user and content level:

- Audit Trail - provides a snapshot of the history of content changes;
- Version Control - automatic backing up of content as changes are made.

5. **Maintain Consistency**

- Presentation Consistency - content should be presented within pre-defined templates whenever possible;
- Brand Integrity - enforced by limiting available logos and layout design in a manner consistent with corporate policies.

A comparison in realizing the business processes specific to content management with a CMS vs. without a CMS is shown in the following table [GS-06][HH-10]:

<b>Business Process</b>	<b>With a CMS</b>	<b>Without a CMS</b>
<b>New Page Creation</b>	A new page is created based on a pre-defined default. All navigation links are automatically updated and a full audit trail is available.	A new page is created as a copy of an existing one. The site map and context navigation links must be updated by hand and standards enforced in an ad-hoc manner.
<b>Content Consistency</b>	Templates are separated from page content, strictly maintaining consistency throughout the site. Display consistency is enforced by the CMS.	Content and template are inextricably tied together, making it difficult to update changes site-wide. Display consistency is determined by the developers.
<b>Workflow Processes</b>	Workflows are built to mirror designated business processes. The CMS workflow engine records an audit with comments on each step. Upon final approval, content is automatically published online.	Workflow is typically done via email in an ad-hoc fashion. Emails are sent to different persons in the organization and upon subsequent approvals, manually published online.
<b>Publishing Times</b>	Content is published immediately once necessary approvals have been made.	Content is published when the webmaster has available time, which could take several days and incur re-configuration errors.
<b>Legal Compliance</b>	Compliance is enforced by the system maintaining records of content changes and content publication.	Compliance is left up to the team members. Changes to the content must be manually backed up and a log kept of when content was published.

The adoption of Web CMS solution by an organization permit to that to achieve the following specific goals:

- Streamline and automate content administration and workflow management;
- Increase the formalization and easiest the administration by adoption of Web-forms-based content creation and administration and allows for automatic acquisition of associated metadata;
- Creates the environment ensuring distribute content creation, management, and control;

- Separate the content from layout and design and redirects the responsibility for content creation and publication to the content contributors (authors);
- Creates database oriented reusable content repositories;
- Allow imposing methods and sophisticated rules for access control and security;
- Allows for customization of user and application interfaces ensuring, at least in that way, even the integration with legacy systems;
- Allows for scheduled automatic content archiving and for version control.

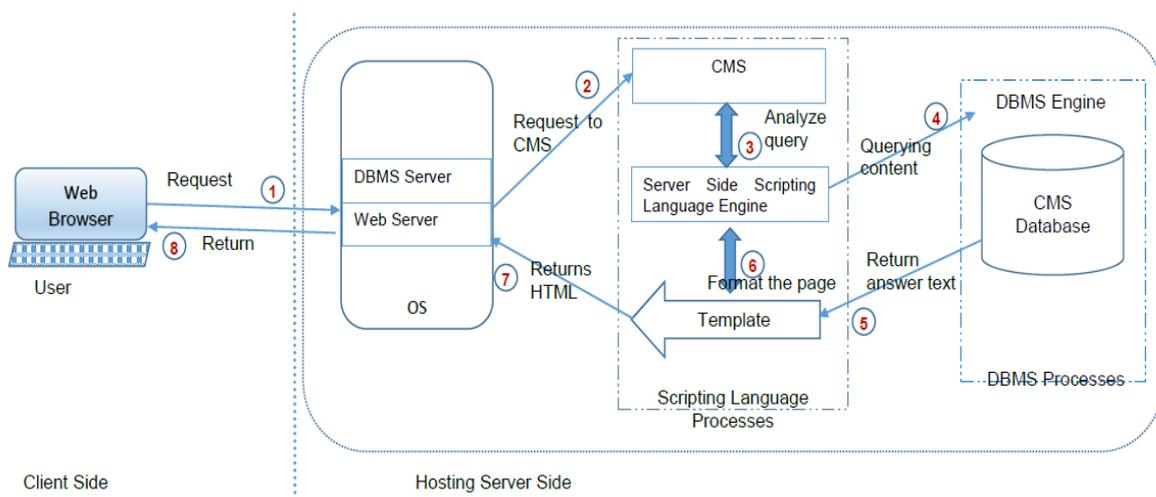
### 6.3 Open Source CMS Architecture and Functionality

The Figure 1 illustrates the way Web server, on which the CMS solution is hosted, answers to the user request and shows the major infrastructure components and processes that contribute to realizing the solution.

The CMS solution is based on an infrastructure that must exist at hosting server side containing at least the following components:

- The host operating system (as Linux is, for example);
- The Web server application (such as Apache) configured to be aware about all other components (described down here);
- The server side scripting/ programming language interpreter in which CMS described (PHP in most open source implementations, Java, C# etc.);
- The database management server (for most open source solutions is MySQL);
- The CMS application deployed in a Web domain hosted by the Web server.

The web browser requests a web page (1) maintained by a CMS application hosted in a domain on a Web server. The request for the web page is realized in the regular way the pages requested on web (by specifying its URL/ URI address). The web server address the request to the CMS application (2). A part of CMS is loaded into the web server and (3) starts execute into the server side scripting language used to implement the CMS application (usually PHP) engine and analyzes the request to determine the requested content. The CMS engine establish a connection to the database via DBMS engine and sends



**Figure 1 Open Source CMS Architecture and Functionality**

the request (expressed as a SQL query) for the specified article (4). Once the article retrieved the DBMS engine sends (5) it as a formatted text string (as unformatted series of strings corresponding to article parts such as article title, article body, etc.) to the CMS which in turn formats this (6) as specified in the style of the user template. CMS creates the HTML answer and transmit this (7) to the Web server which in turn downloads (8) that page to the user browser.

Modern CMSs typically have workflow processes that start at content creation and move through editing or approval stages to publishing. They have the capability to extend site functionality, to a broad category of specific needs, by the ability to use a wide range of extensions or add-ons. For example,

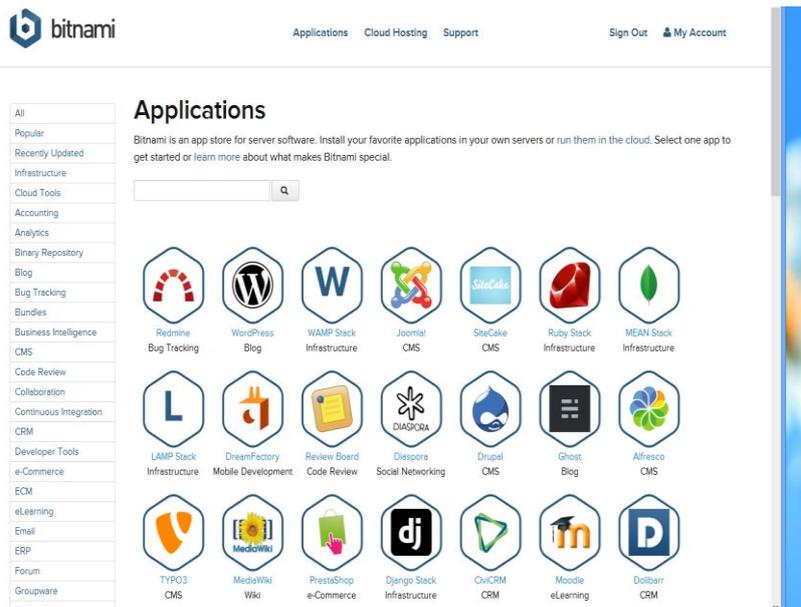
Joomla has more than 4,000 extensions available through various forums and newsletters or to the official website <http://extensions.joomla.org>.

## 6.4 Setup and Installing Locally Open Source CMS Solutions

### Setup WAMP stack

The steps explained below makes reference to the open source products available as installable stacks/ modules at <http://www.bitnami.com> (Figure 2) for different platforms.

The installation here realized by adopting the module installation type allowing to install all desired solutions as a stack atop to the infrastructure on a Microsoft Windows 8 on 64 bits machine.



**Figure 2 Bitnami.com Applications Page**

When accessing an application by clicking his associated icon or the blue name the page allowing downloading for a specific platform the user provided in that page with a brief description and with a link to the official websites of the groups/ companies that create, maintain and deliver the solution. Because most of the time the downloadable packages do not contains the full set of product documentation and related articles we must find all these in the product official website.

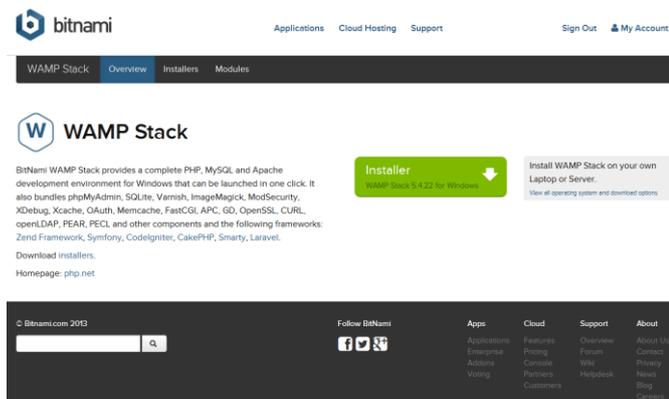
After the infrastructure installed the computer transformed into a locally Web server. By placing the websites, other than those corresponding to an open source application installable atop all previous installed Bitnami modules, in the Bitnami installation folder in the subfolder *apache2/htdocs* the website will be accessible via HTTP server by invoking its “local” URL, for example <http://localhost/avrams/html5/html5-example-web-page.html>. The figure below shows the structure of the Bitnami folders for that example URL.

All the figures here are screen captures during the setup, installation, and usage of the Bitnami WAMP stack.

**Step 1.** Access the Bitnami.com Applications Page (Figure 2)

The products offered here are:

- Independent infrastructures stacks available for Windows (WAMP, for example), Linux (LAMP, for example), or Mac OS (MAMP, for example);
- Product stacks containing both the infrastructure and the product itself.



**Figure 3 WAMP Stack Installer**

**Step 2.** Choose the wanted infrastructure stack (Figure 3; Figure 4)

Here the WAMP stack for Windows containing the Apache web server, MySQL database management system together with its usual administrative application phpMyAdmin, PHP environment, and a set of development frameworks. You can choose the proper version (Figure 4) that fits to the requirements of other applications you installed on the same machine.

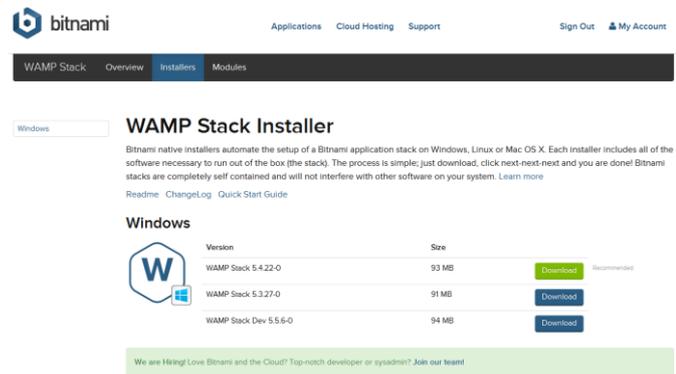
**Step 3.** Start the downloaded application executable (Figure 4) (is better in Windows 7 and following to lunch the application as Administrator (Run as Administrator)).

The application setup wizard will guide you to the entire installation process. The parameters you must specify here represented by the name and location (pathname) of the installation folder and the password for the MySQL username called “root” (is the MySQL Administrator). This password will be required to build the connection string for all applications using this MySQL server (including to open the phpMyAdmin database administrative console).

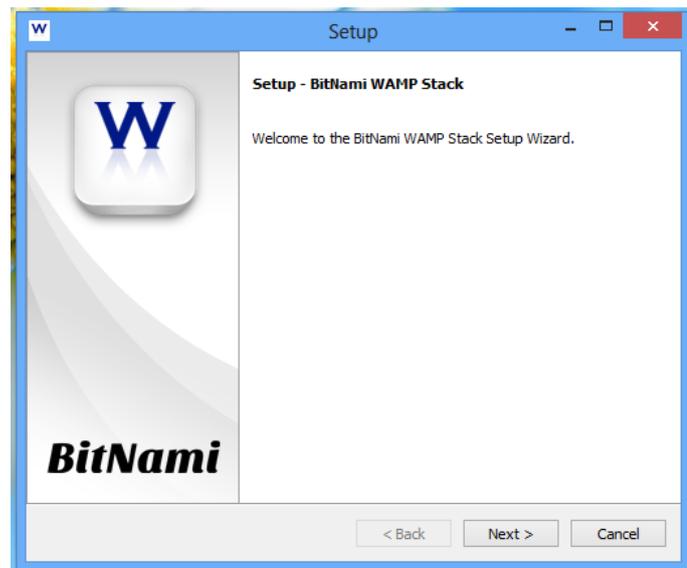
The phpMyAdmin is the only imposed framework installed without the express specification of the user because that option realized when the infrastructure stack chosen.

**Step 4.** Choose the frameworks to be installed (Figure 6)

In the next screen the user can choose which of the available development framework will be installed (Figure 6):

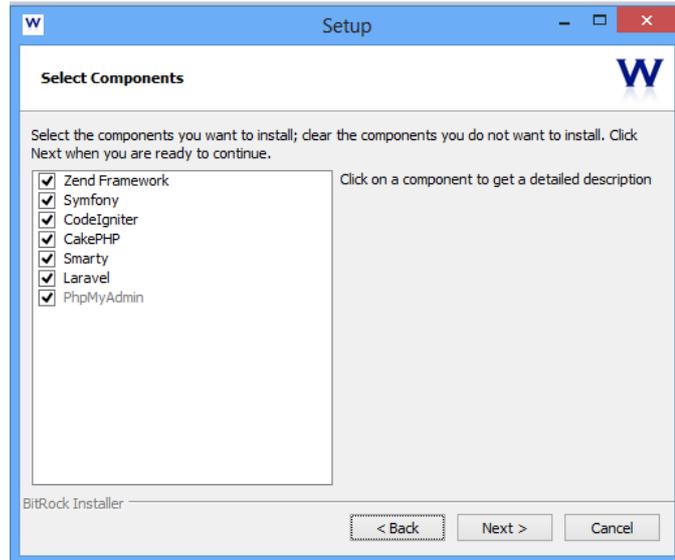


**Figure 4 WAMP Stack Installers**

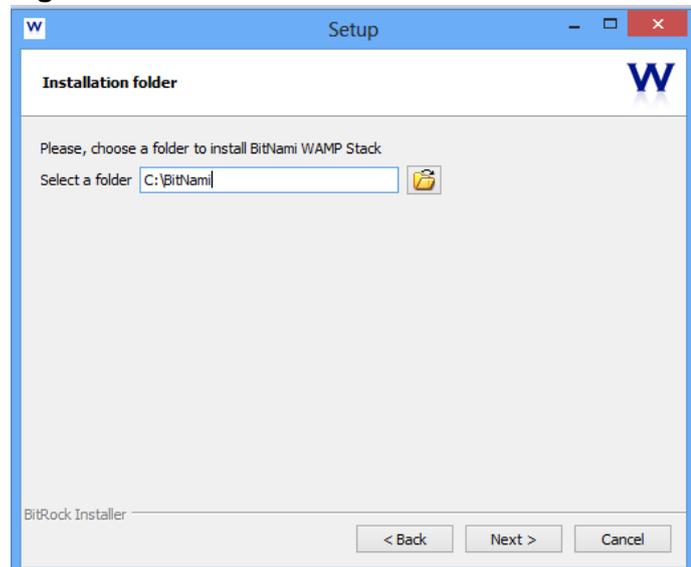


**Figure 5 Bitnami WAMP Stack - Setup**

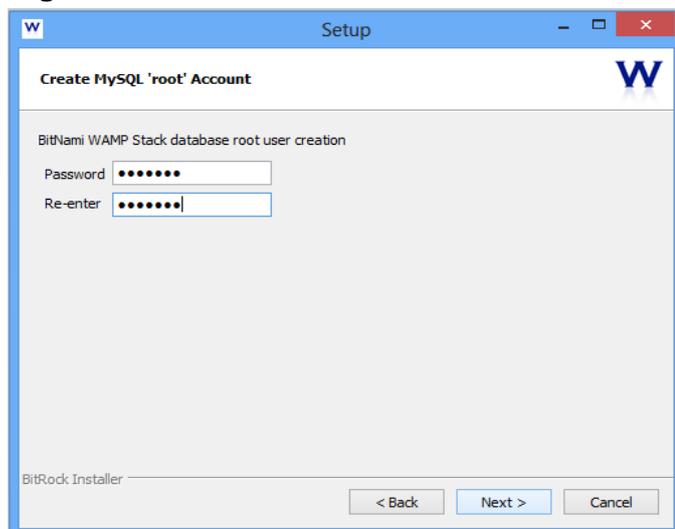
- Zend Framework - is an open source framework, implemented using 100% object-oriented code, for developing web applications and services with PHP 5;
- Symfony – a PHP framework;
- CodeIgniter - is a powerful PHP framework with a very small footprint, built for PHP coders who need a simple and elegant toolkit to create full-featured web applications;
- CakePHP - it's a foundational structure for programmers to create web applications in a structured and rapid manner—without loss of flexibility;
- Smarty - is a template engine for PHP that facilitates a manageable way to separate application logic and content from its presentation;
- Laravel - is a web application framework with expressive syntax.



**Figure 6 Bitnami WAMP Stack - Frameworks**



**Figure 7 Bitnami WAMP Stack - Folder**



**Figure 8 Bitnami WAMP Stack – MySQL Administrator**

**Step 5.** Specification of the installation drive and folder (Figure 7);

**Step 6.** Specify the password for the MySQL server Administrator - the “root” account (Figure 8);

**Step 7.** The installation process (Figure 9) is realized automatically and consist in unpacking, configuring, and installing in the hard drive the components of the stack.

It builds also the application management (called *manager-windows.exe*, and located in the folder where Bitnami stack installed) console (Figure 10) that can be used to configure and manage the servers (Manage Servers tab – Figure 11) and to access the applications installed (Go To Application – Figure 12).

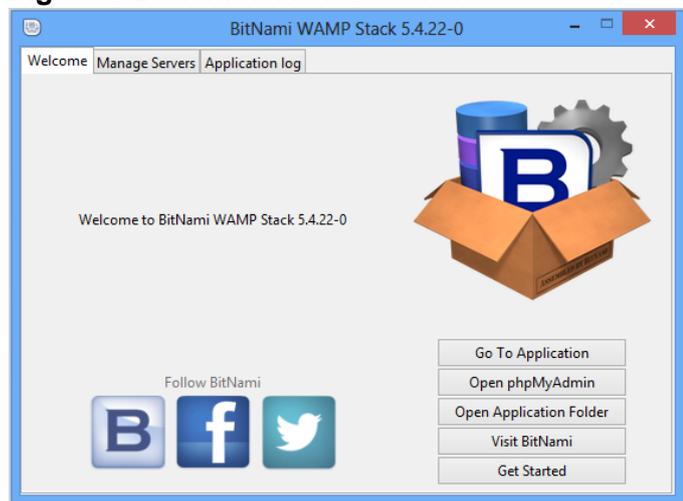
When the button Configure for a server is selected the application open a dialog box allowing the visualization and to analyze the server logs (events or errors) or to access the configuration file of the server (Figure 11) via the text editor available in the Windows environment (*Notepad.exe* for most cases).

By pressing *Go To Application* command button in the management application in the *Default* browser set up in the system is started the web application pointed by *localhost* (local IP address 127.0.0.1). The application is declared in Apache configuration file by its configuration file: Include "C:/BitNami/apache2/conf/bitnami/bitnami.conf"

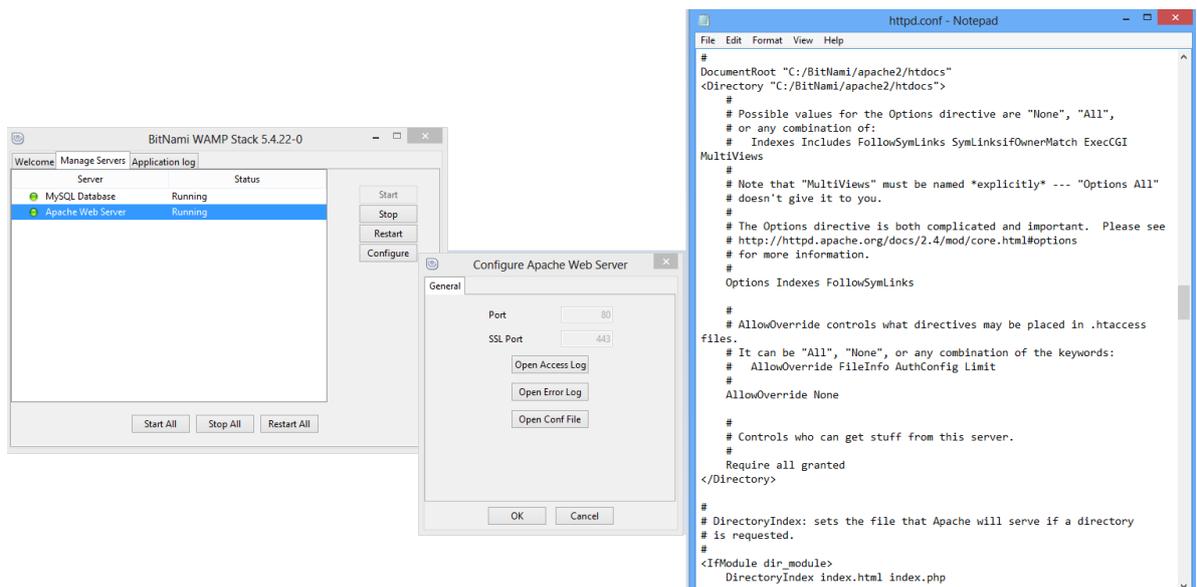
which contains the definition of the location of the application main web page (the local address):



**Figure 9 Bitnami WAMP Stack - Install**



**Figure 10 Bitnami WAMP Stack – Management Application**



**Figure 11 Bitnami Management Application – Access to Server Configuration File (Apache here)**

```

<VirtualHost _default_:80>
  DocumentRoot "C:/BitNami/apache2/htdocs"
  <Directory "C:/BitNami/apache2/htdocs">
    Options Indexes FollowSymLinks
    AllowOverride All
    <IfVersion < 2.3 >
      Order allow,deny
      Allow from all
    </IfVersion>
    <IfVersion >= 2.3 >
      Require all granted
    </IfVersion>
  </Directory>

  # BitNami applications installed with a prefix URL (default)
  Include "C:/BitNami/apache2/conf/bitnami/bitnami-apps-prefix.conf"

</VirtualHost>

```

The default browser displays the page containing two tabs:

- **Welcome** – containing useful links to information regarding the operation of the application (to verify/ check, to add new modules etc.);
- **Applications** – giving you a brief description of every application installed via **Bitnami** installers and access to that application. For other applications you deploy in the *htdocs* folder of Apache webserver and that you configure as local domain name or to which you associate a starting port number of the *localhost* you can change manually the page *applications.html* located in *<Bitnami installation folder>/apache2/htdocs/applications.html*, by adding the desired entries. These changes will be kept as such even you add/ remove modules using Bitnami setup files.

Figure 13 shows the page Applications after the installation of the Bitnami infrastructure stack and which allows access only to administer the MySQL database.

By clicking the Access hyperlink this will display the *phpMyAdmin* login window where we must logon for the first time as username “root” and with the password we provide during setup process (Figure 14).

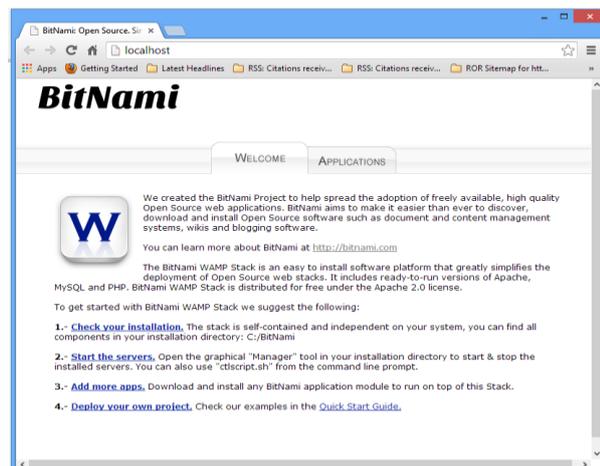


Figure 12 Bitnami WAMP Stack - Applications

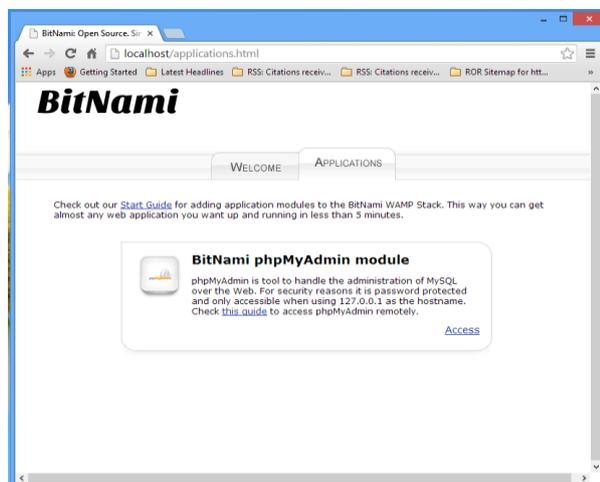
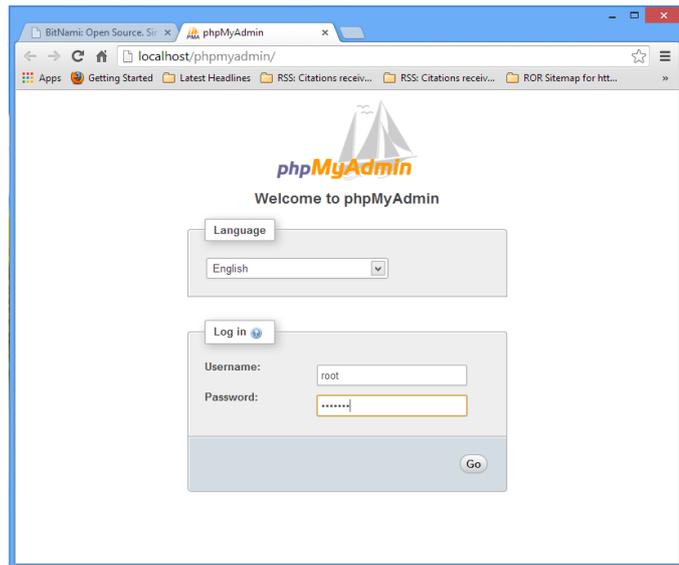


Figure 13 Bitnami WAMP Stack - phpMyAdmin

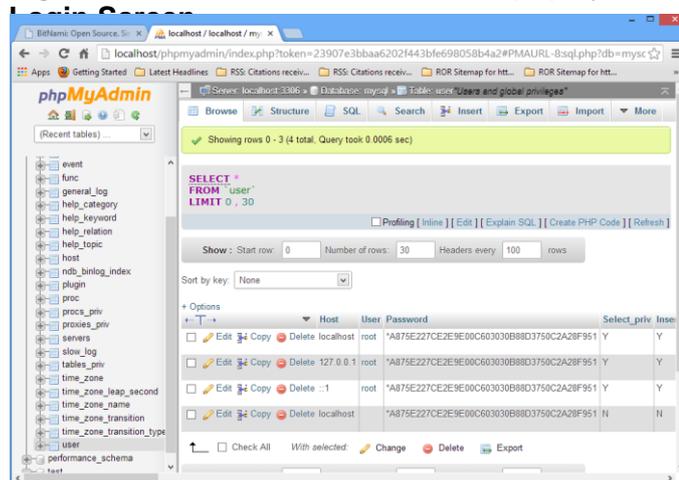
The MySQL database server verifies the validity of the combination username/password you supply and if checked displays the database management console depicted in Figure 15, as a web application in your default browser.

The database management module allows the entire set of SQL DBA and SQL DML languages that permit to: define/ erase databases, create/ destroy tables, fill/ insert data into tables, update, delete, rows, browsing, defining queries and views etc. It allows also to export/ import data in different formats, to realize total/ partly backups, to realize a complete management databases and their users.

For the used version you must see the used syntaxes for the SQL sentences since they can differ easily from one version to another depending on which version of SQL language specification is implemented. The queries, and also almost of the realized operations on the databases can be saved as PHP sequences which in turn can be included in the pages you want use them.



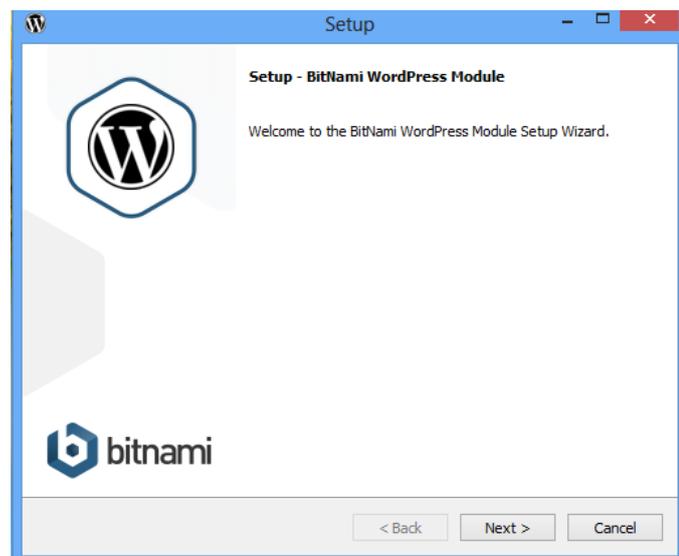
**Figure 14 Bitnami WAMP Stack – phpMyAdmin**



**Figure 15 Bitnami WAMP Stack – Databases Management**

## Setup WordPress Module

The setup process is started by double clicking the module installation executable. In the following screen are illustrated the main screens of the setup process for the most used open source application for blog management WordPress (Figure 16).



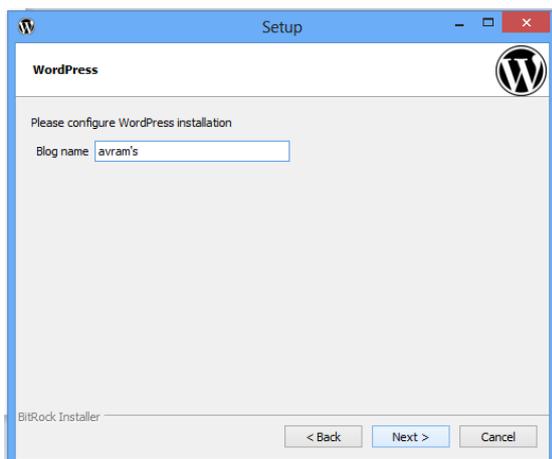
**Figure 16 Bitnami WordPress Module Setup**

The Setup process request the user to specify the folder in which the module will be installed (Figure 17). For an easy of management is better be the same folder as the infrastructure stack.

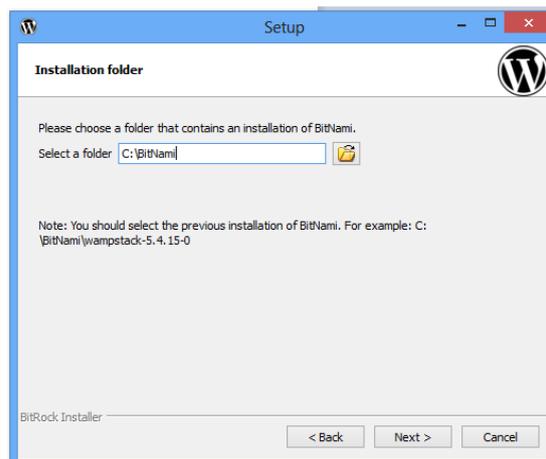
The module has two parts:

- A shared part that is available for all website/blog visitors;
- An administrative console that is available only for users granted with that right. In the setup process we must supply to that process the identification data for the administrator account (Figure 18): login name, real name, email address where can be contacted, and the password for the MySQL database management system required to add that user and to create the database container for WordPress application.

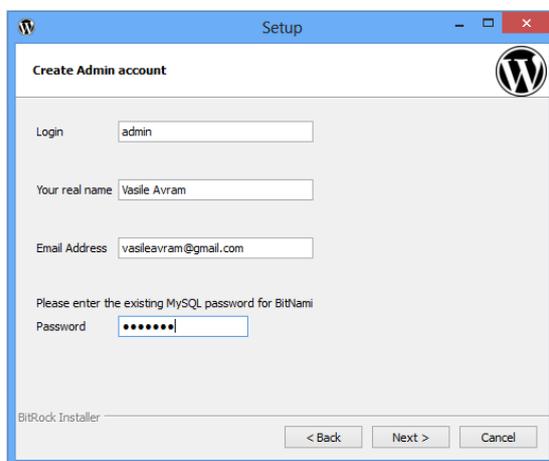
After the Admin account created we must specify the name chosen for the blog (Figure 19) that will appear on the heading part of the pages. The next screen give the possibility to specify if created a connection with the mail system provider chosen.



**Figure 19 Bitnami WordPress Module Setup – Blog Given Name**



**Figure 17 Bitnami WordPress Module Folder**



**Figure 18 Bitnami WordPress Module Setup – Administrator Account**



**Figure 20 Bitnami WordPress Module Install**

The installation process start and add, to the existing installed modules, WordPress as a new application provided with all required resources. It modifies also the stack management application and the main page, tab Applications (Figure 21), of the web page allowing starting the application from the web page. The application will include two modules: one available to all blog visitors (Figure 22), including administrators, and another accessible only to authenticated users (Figure 24), as the administrators are, a dashboard (Figure 25) allowing the management of themes, customizing, approval of comments, adding/ erasing articles and menus, archiving etc. Figure 23 shows the blog in Figure 22 after customization using the available tools in the dashboard and by changing the chosen theme code in the files:

- a) <http://avrams.eu/wp-content/themes/sprachkonstrukt2/style.css>
- b) <http://avrams.eu/wp-content/themes/sprachkonstrukt2/js/modernizr.custom.23499.js>

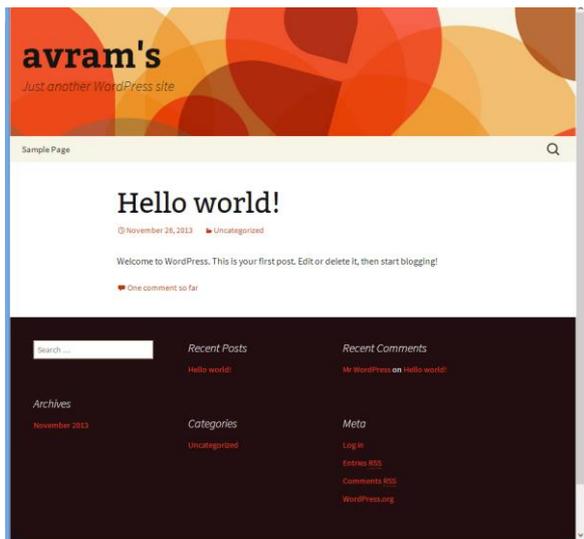


Figure 23 WordPress – Blog Main Page (not yet customized)

After authentication the administrative dashboard (Figure 25) is displayed to the screen and allows access to a complete set of operations to customize, define, update, delete and/ or archive.

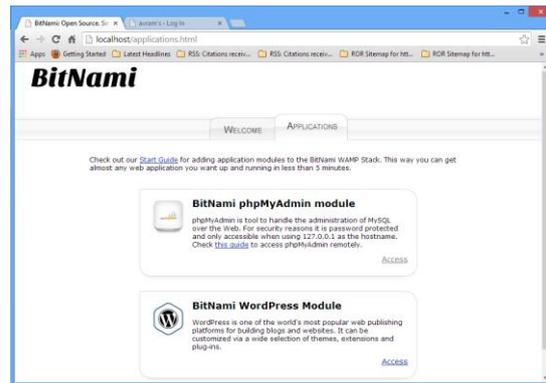


Figure 21 BitNami WordPress Module Install

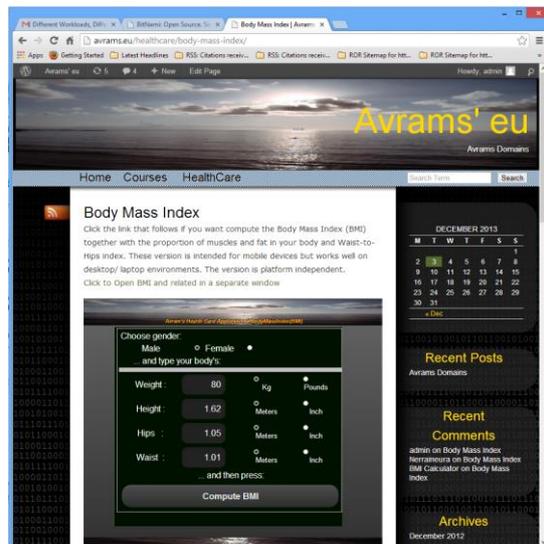


Figure 22 WordPress – Blog Main Page (not yet customized)



Figure 24 WordPress – Access Dashboard

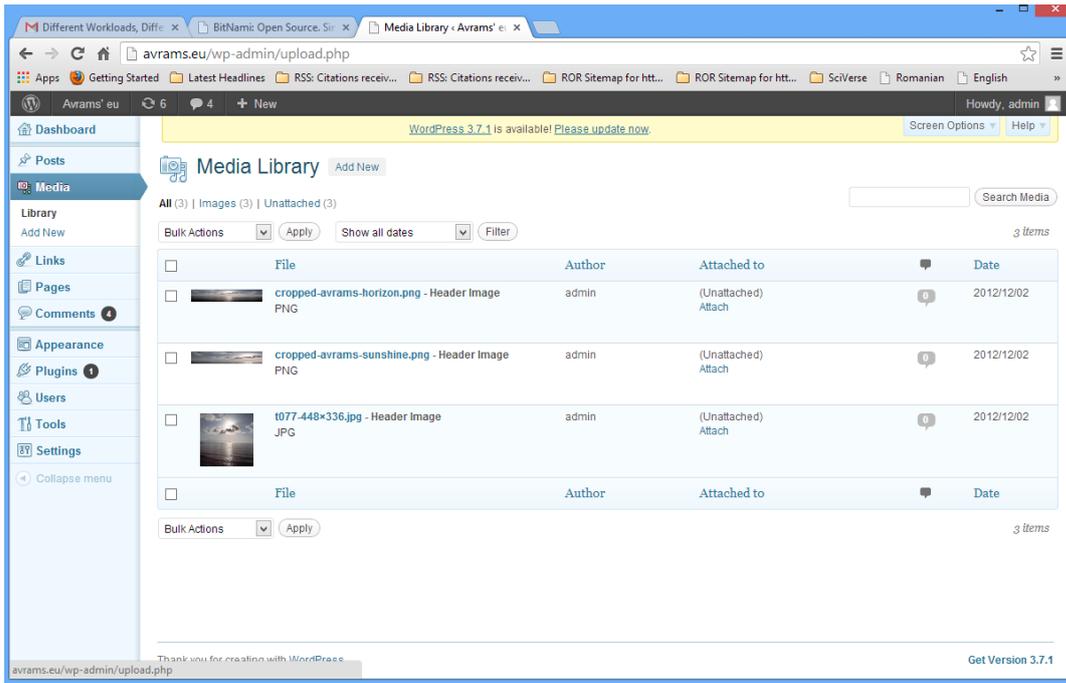


Figure 25 WordPress Module Install

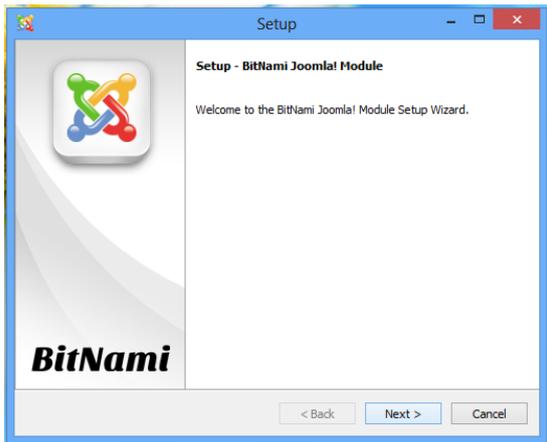


Figure 27 Bitnami Joomla - Module Install

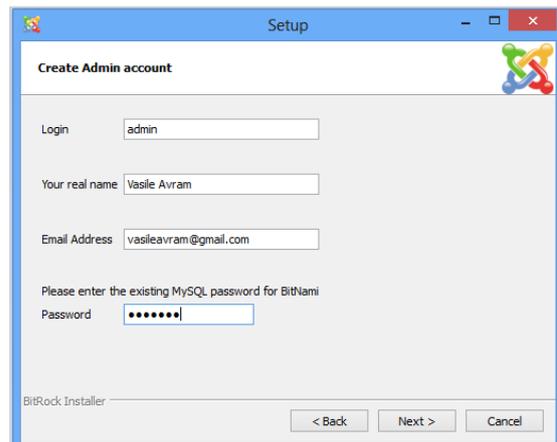


Figure 26 Bitnami Joomla – Create Admin Account

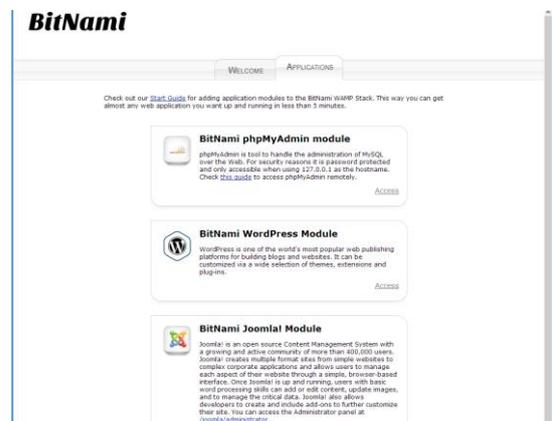


Figure 29 Bitnami Joomla – Applications Entry Page



Figure 28 Bitnami Joomla – Install

## Setup Joomla Module

The Joomla module is the most used CMS open source solution. The module will be installed atop the existing applications in the Bitnami folder.



Figure 31 Bitnami Joomla – Create Admin Account

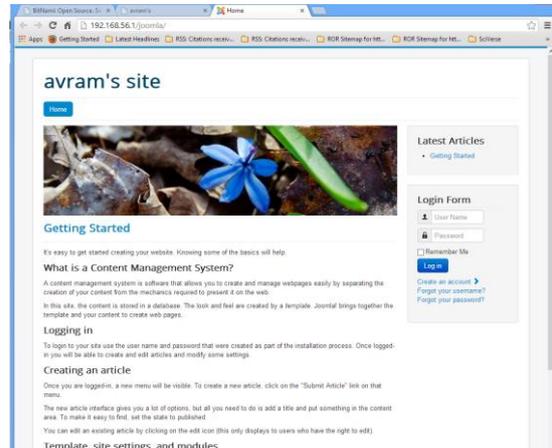


Figure 30 Bitnami Joomla – Generated Site

The setup/ install process is similar with the one for the WordPress module.

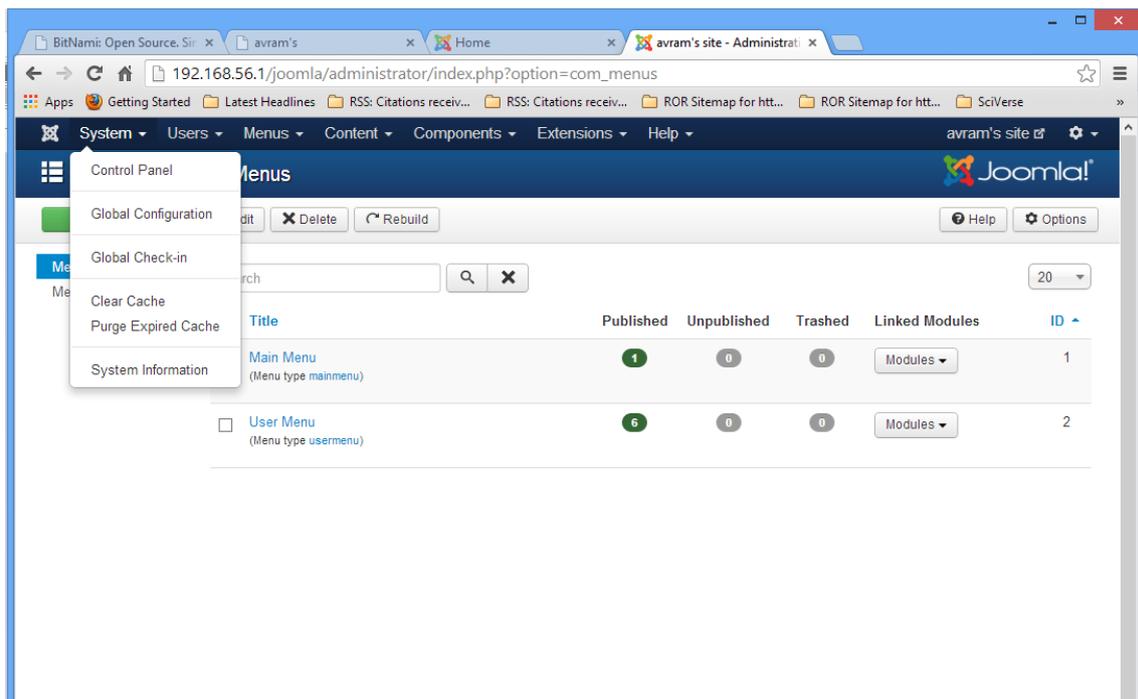


Figure 32 Bitnami Joomla – Administrative Console

## Annex 1. Sample CMS ROI Calculator [Gs-06]

<b>Variable Factors</b>	
How many departments, divisions, and subsidiaries are in your organization?	10
How many pages of content does each one create per month?	10
How many pages of existing content does each one update per month?	10
How many documents does each one publish per month?	10
How many times are the navigation menus manually updated per month?	15
In how many formats is the content published (printer friendly, WML, etc)?	2
How much does it cost per hour for web developers/consultants?	\$75.00
<b>Time Assumptions Without a CMS</b>	
How many extra minutes are required to put a new page online?	20
How many extra minutes are required to update an existing page?	10
How many extra minutes are required to publish and link to a file?	5
How many extra minutes are required to update navigation menus?	10
<b>Monthly Costs Without a CMS</b>	
Create Content	\$2,500.00
Update Content	\$1,250.00
Publish Documents	\$625.00
Update Navigation Menus	\$187.50
Reformat Content	\$2,281.25
Monthly Total	\$6,843.75
<b>Total Costs and Savings</b>	
CMS Cost First Year	\$50,000.00
CMS Cost Subsequent Years	\$8,000.00
Total Yearly Costs Without a CMS	\$82,125.00
First Year Savings	\$32,125.00
Subsequent Yearly Savings	\$74,125.00
<b>Return on Investment</b>	
One Year	85%
Two Year	242%
Three Year	340%

## References

[RD-09] Dan Rahmel, Beginning Joomla!, Second Edition, Apress, April 27, 2009, ISBN-10: 1-4302-1642-5, ISBN-13: 978-1-4302-1642-1, Pages in Print Edition: 447

[HH-10] Introduction to Content Management Systems, Hannon Hill Corporation, (2010), <http://www.hannonhill.com>

[GS-06] GlobalSCAPE – Content Management Systems Buyer’s Guide - A comprehensive guide to buying a CMS—what to look for, and what to expect, 2006, <http://globalscape.com>

[Bitnami] <http://www.bitnami.com>

[Zend Framework] <http://framework.zend.com/manual/1.11/en/manual.html>, retrieved on 2 of December, 2013

[CakePHP] [http://book.cakephp.org/2.0/\\_downloads/en/CakePHPCookbook.pdf](http://book.cakephp.org/2.0/_downloads/en/CakePHPCookbook.pdf), retrieved on 2 of December, 2013

[Symfony] <http://symfony.com/doc/current/index.html>, retrieved on 2 of December, 2013

[Smarty] <http://www.smarty.net/documentation>, retrieved on 2 of December, 2013

[Laravel] <http://laravel.com/docs>, retrieved on 2 of December, 2013