

Табельщик имеет права на добавление и удаление данных, доступ к корректировке имеющейся информации, если изменяемая информация относится к текущему месяцу. За предыдущие периоды таблицы уже сформированы и изменить их уже нельзя. В программе предусмотрена возможность вывода сформированных таблиц на печать, это может быть как плановый, так и фактический табель. Также табельщик имеет возможность получать различного рода статистическую информацию из системы. В данной базе имеется широкий спектр функций, таких как связанные запросы, связь с внешними таблицами и базами данных.

Минимизирована необходимость ручного ввода информации, ее необходимо корректировать только в случае, если имеются отклонения в работе от планового табеля.

Е.В. Мельников (УО «ГГУ имени Ф. Скорины», Гомель)
Науч. рук. **В.А. Дробышевский**, старший преподаватель

ПРОЕКТИРОВАНИЕ ВЕБ-САЙТА ДЛЯ ГОМЕЛЬСКОГО ГОРОДСКОГО НАУЧНО-ПРОСВЕТИТЕЛЬСКОГО ОБЩЕСТВЕННОГО ОБЪЕДИНЕНИЯ «ОРАКУЛ»

При разработке сайта за основу архитектуры был взят шаблон проектирования MVC. Этот шаблон предполагает разделение приложения на различные компоненты. Однако веб-сайт не является точной копией классического описания шаблона MVC, а реализует лишь основную идею. Получилась следующая структура приложения (рисунок 1).

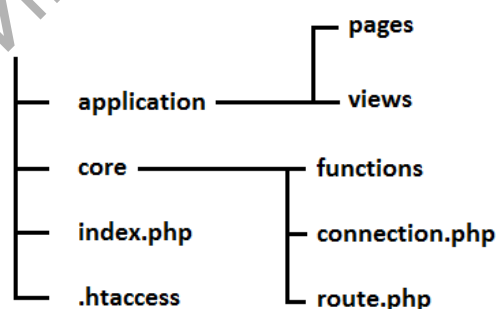


Рисунок 1 – Структура веб-сайта

В каталоге “application” находятся файлы, которые отвечают за то, что будет находиться на сайте, и как это будет выглядеть. В каталоге “core” находятся файлы, которые отвечают за функциональность приложения, независимо от того, какой контент будет отображен на сайте. Файл “index.php” играет роль входной точки всего приложения. Файл “.htaccess” – это файл-конфигуратор apache-серверов, который дает

возможность конфигурировать работу сервера в отдельных папках, не предоставляя доступа к главному конфигурационному файлу. В каталоге “pages” находятся подкаталоги с названием страниц, которые есть на сайте. В каждом из таких подкаталогов содержится РНР файл, который генерирует контент и подключает соответствующий вид из папки “view”. В каталоге view содержатся php файлы, содержащие html разметку с php вставками для вывода содержимого определённой переменной. В этой папке находятся css стили и javascript файлы. В каталоге functions содержатся файлы с php функциями, которые используются в других частях приложения. Файл “connection.php” содержит константы для конфигурации подключения к базе данных. Файл “route.php” играет роль маршрутизатора. Он разбирает URL запрос и подключает требуемый php файл из папки pages.

Е.В. Мельников (УО «ГГУ имени Ф. Скорины», Гомель)
Науч. рук. **В.А. Дробышевский**, старший преподаватель

ПРОГРАММНАЯ ЛОГИКА ВЕБ-САЙТА ДЛЯ ГОМЕЛЬСКОГО ГОРОДСКОГО НАУЧНО-ПРОСВЕТИТЕЛЬСКОГО ОБЩЕСТВЕННОГО ОБЪЕДИНЕНИЯ «ОРАКУЛ»

При разработке веб-сайта было решено использовать процедурный стиль вместо ОПП. Поскольку данный проект имеет довольно простую структуру, то было решено пренебречь такими особенностями ООП как удобная командная разработка и масштабирование (что необходимо крупным проектам) в пользу производительности. Однако это не означает, что весь код собран в одной куче. Принципы MVC соблюдены. Далее описывается, как работает данный проект. Сначала все запросы поступают на файл index.php, который находится в корне проекта. В нём определяются главные константы и route.php:

```
<?php
$root = getcwd() . '/';
define('APP', $root . 'application/');
define('CORE', $root . 'core/');
define('PAGES', APP . 'pages');
define('VIEWS', APP . 'views/');
define('VIEWS_REL', '/application/views/');
define('FUNCTIONS', APP . 'functions/');
include CORE.'route.php';

php?>
```

Далее управление переходит в route.php. Там подключаются файл functions.php, который содержит необходимые для работы приложения функции. Затем в этом файле разбирается URL на составные части. Затем проверяется, есть ли папка с названием, которое указано в URL