

## Описание задания для Демонстрационного экзамена

Задание состоит из четырех обязательных (на 2 полных дня) и одного дополнительного модулей (пятый модуль (WordPress) добавляется, если предусмотрен третий день соревнований). При этом в течении дня конкурсант выполняет одно задание, состоящее из двух модулей. Нарушать порядок выполнения модулей в рамках одного дня нельзя.

Задания построены с учетом WSSS:

- Организация работы и управление (WSSS 6%)
- Коммуникационные и межличностные навыки (WSSS 6%)
- Графический дизайн веб-страниц (WSSS 22%)
- Верстка страниц (WSSS 22%)
- Программирование на стороне клиента (WSSS 22%)
- Программирование на стороне сервера (WSSS 14%)
- Системы управления контентом (Content management systems) (WSSS 8%)

### День 1 (Дизайн и верстка + JavaScript + PHP)

Требуется разработать онлайн игру «Space pipes».

#### Сюжет:

*При столкновении с космическим телом на Международной космической станции «Марс-2» в секторе «А» произошел процесс неконтролируемой разгерметизации.*

*Требуется срочно восстановить разрушенную систему подачи воздуха. Для этого в сектор «А» были направлены специалисты по монтажу систем вентиляции. Перед ними поставлена задача, восстановить систему воздуховода на поврежденном участке, соединив нужные участки труб между началом и концом разрыва (начало и конец разрыва находятся на противоположных сторонах).*

#### Общие требования:

Игра имеет три уровня сложности, отличающихся размерами игрового поля. Начало и конец трубы находятся в верхнем левом и нижнем правом углах соответственно. Клетки поля заполняются случайным образом изображениями фрагментов трубы только в том случае, если существует хотя бы один путь. При каждом щелчке игрока по изображению фрагмент поворачивается на 90 градусов. Игра заканчивается либо с построением правильного пути, либо с окончанием времени таймера.

При реализации игры необходимо предусмотреть три варианта экрана:

- стартовый экран - содержит наименование игры, уровень сложности игры (новичок, продвинутый, мастер), поле ввода своего имени, а также кнопку “Начать игру”. Кнопка “Начать игру” активна, если введено имя пользователя и выбрана сложность.
- экран игры - интерфейс должен содержать:
  - блок с таймером - содержит таймер обратного отсчета с указанием минут, секунд в формате мм:сс. В этом же блоке располагается кнопка “пауза”, останавливающая таймер и блокирующая изменение игрового поля.
  - блок с заработанными баллами (количество оставшихся секунд по таймеру) - количество заработанных игроком баллов за текущую игру.
  - блок с именем игрока - если игрок на стартовом экране в имени ввел “tester”, то имя пользователя должно отображаться по-другому, так как это “кодовое” слово для игры в тестовом режиме.
  - блок с руководством по игре.

- размер поля генерируется в зависимости от сложности игры:
  - Новичок – размер поля 4x4;
  - Продвинутый – размер поля 5x6;
  - Мастер – размер поля 6x6;
- экран окончания игры - содержит таблицу результатов с первыми 10 лучшими результатами (имя, уровень сложности, баллы). Если игрок по баллам не вошел в 10 лучших, то вместо 10 места показывается его результат с указанием места в таблице рекордов.

Участнику необходимо разработать дизайн, реализовать указанный функционал игры, а также разработать понятный и удобный интерфейс игры (менять игровую логику запрещено). Реализуемая анимация должна отражать сюжет игры и побуждать игрока к действиям.

HTML и CSS код должны быть валидными. Создаваемые файлы должны структурированы и содержать комментарии. PHP и JS код должен выполняться без отображения ошибок, в т.ч. в консоли браузера.

Браузером для проверки основных функциональных возможностей является Mozilla Firefox Developer Edition. Однако работа приложения будет также проверена в браузере Google Chrome для проверки кроссбраузерности программы.

## **Модуль 1 (3 часа)**

В данном модуле необходимо решить следующие задачи:

- Разработка дизайна и элементов интерфейса игры, учитывая тематику, логику и удобство использования интерфейса.
- Верстка стартового экрана, игрового поля (заполненное элементами, в зависимости от уровня сложности) и экрана окончания игры (с таблицей рекордов).
- Обработать изображения в соответствии с размерами блоков.
- Анимированные экраны: стартовый и окончания игры.
- Таймер обратного отсчета - начинает обратный отсчет с началом игры (1 минута), как только доходит до значения 00:00 игра заканчивается.
- Реализована БД с тестовыми записями для вывода в таблицу рекордов по уровням сложности.

Выходные файлы: CSS, HTML, JS.

## Модуль 2 (3 часа)

Реализовать логику, в состав которой должны быть включены следующие функции:

- Алгоритм заполнения полей для разных уровней сложности.
- Алгоритм определения правильного пути и окончания игры.
- Пауза игрового процесса - останавливается время на таймере, запрещается воздействие на игровое поле. Режим паузы также может быть инициирован по нажатию на клавишу пробел. Возобновление игры так же возможно по нажатию на клавишу “пробел” или по нажатию на кнопку “Пауза” на игровом поле.
- Реализована таблица рекордов. Данные в таблицу рекордов передаются без обновления страницы.
- Сохранение итоговых результатов в базу данных на стороне сервера.
- Режим теста – режим, в котором таймер обратного отсчета не запускается, и игра не останавливается при взаимодействиях, которые подразумевают проигрыш или конец игры. Результаты не записываются в базу данных.

Игровой процесс должен сопровождаться подсказками, необходимо разработать краткое руководство по игре.

Разрешенные JS-библиотеки: jQuery, jQuery UI. JavaScript фреймворки запрещены.

Выходные файлы: CSS, HTML, JS, PHP.

В приведенной ниже таблице описывается, как распределяются баллы относительно разделов WSSS:

## Задание второго дня (PHP + JavaScript)

Вам необходимо разработать сервис записи на ремонт космических станций. Целевой аудиторией сервиса являются специалисты: Роскосмоса, NASA и CNSA.

Общие требования к сервису:

- 1) Для реализации сервиса используется предоставленный шаблон (находится в каталоге service, шаблон администратора находится в каталоге service /admin);
- 2) Все файлы сервиса структурированы по каталогам, наименования файлов директорий должны отражать их содержание;
- 3) Допустимые символы в названии файлов: символы латинского алфавита, цифры 0-9, символ нижнего подчеркивания;
- 4) Сайт доступен по ссылке <http://xxx/service>;
- 5) Панель управления сайтом – страница доступная по адресу <http://xxx/service/admin>;
- 6) Данные для входа администратора: логин – admin, пароль – admin;
- 7) Вся валидация должна происходить на стороне сервера;
- 8) Все ошибки должны выдаваться в виде информативных сообщений пользователю;
- 9) Код должен быть структурирован и содержать необходимые комментарии;
- 10) После авторизации пользователя на страницах сервиса должны присутствовать: имя пользователя, аватар;
- 11) HTML и CSS код должны быть валидными. Работа будет проверяться в браузере Google Chrome.

Использование PHP Frameworks по желанию участника. Также разрешено использование JQuery, JQuery UI.

### Модуль 3 (3 часа)

#### 1. Сущность «Зарегистрированный Пользователь»

##### 1.1 Глоссарий атрибутов

Атрибут	Условия	Обязательность для заполнения
ФИО	без сокращения, содержит только кириллицу, без цифр и знаков препинания	да
E-mail	валидируется на соответствие шаблону e-mail адресов	да
Номер телефона	В формате +X(XXX)XXX-XX-XX	нет
Логин	должен быть уникальным, минимум 6 символов, символы латиницы	да
Пароль	должен содержать не менее 6 символов английской раскладки, верхнего и нижнего регистра	да
Подтверждение пароля	должно совпадать с полем «Пароль»	да
Изображение пользователя (аватар)	в формате jpg(jpeg) или png с ограничением по размеру не более 1 Мб	нет

##### 1.2 Функции

Наименование	Роль	Пользовательский сценарий	Функциональный сценарий
Регистрация	Гость	<ol style="list-style-type: none"> <li>1. Пользователь вводит значения всех атрибутов</li> <li>2. Если атрибуты введены с ошибкой, то выводится сообщение и выделяются соответствующие поля</li> <li>3. Если всё введено корректно, происходит перенаправление пользователя на страницу авторизации</li> </ol>	<ol style="list-style-type: none"> <li>1. Сервис получает значения атрибутов</li> <li>2. Если валидация не прошла успешно, то сервис возвращает соответствующее сообщение</li> <li>3. Сервис добавляет запись в соответствующую таблицу БД.</li> </ol>
Авторизация	Гость	<ol style="list-style-type: none"> <li>1. Пользователь вводит значения атрибутов логина и пароля</li> <li>2. Если возвращается сообщение об ошибке, то оно должно быть отображено пользователю</li> <li>3. Если логин admin, пароль admin, то пользователь перенаправляется в панель управления сайтом.</li> <li>4. Пользователь перенаправляется в личный кабинет</li> </ol>	<ol style="list-style-type: none"> <li>1. Сервис получает значения атрибутов логина и пароля</li> <li>2. Если логин и пароль не соответствует ни одному пользователю системы, то сервис возвращает соответствующие сообщение об ошибке</li> <li>3. Если логин admin, пароль admin, то пользователю назначается роль Администратора</li> <li>4. Гостю назначается роль Зарегистрированного Пользователя</li> </ol>
Выход	Зарегистрированный Пользователь, Администратор	<ol style="list-style-type: none"> <li>1. Пользователь нажимает на ссылку «Выход»</li> <li>2. Пользователь перенаправляется на страницу авторизации</li> </ol>	<ol style="list-style-type: none"> <li>1. Сервис получает идентификатор пользователя</li> <li>2. Пользователю назначается роль Гостя</li> </ol>

## 2. Сущность «Услуги»

### 2.1 Глоссарий атрибутов

Атрибут	Условия	Обязательность для заполнения
Наименование	Не более 255 символов	Да
Стоимость	>0	Да
Изображение	в формате jpg(jpeg) или png с ограничением по размеру не более 1 Мб	Нет
Количество мест	Число [0..50]	Да
Длительность оказания	Вносится в часах, с возможность указания дробной части	Да
Описание	Текстовое описание без изображений до 1000 символов	Нет
Порядок вывода	Число, определяющее приоритет вывода	Да

## 2.2 Функции

Наименование	Роль	Пользовательский сценарий	Функциональный сценарий
Просмотр	Гость, Зарегистрированный Пользователь, Администратор	<ol style="list-style-type: none"> <li>1. Переходит на страницу просмотра списка услуг</li> <li>2. Пользователь имеет возможность просмотреть все услуги</li> </ol>	<ol style="list-style-type: none"> <li>1. Сервис возвращает полный список услуг, в соответствии с приоритетом</li> </ol>
Добавление услуги	Администратор	<ol style="list-style-type: none"> <li>1. Пользователь вводит значения всех атрибутов</li> <li>2. Если не возвращено сообщение об ошибке, то выводится сообщение об успешном добавлении услуги</li> </ol>	<ol style="list-style-type: none"> <li>1. Сервис получает значения всех атрибутов</li> <li>2. Если валидация не прошла успешно, то выводится сообщение об ошибке</li> <li>3. Добавляет новую услугу</li> </ol>
Редактирование услуги	Администратор	<ol style="list-style-type: none"> <li>1. Пользователь выбирает услугу</li> <li>2. Пользователь вводит изменённые значения необходимых атрибутов</li> <li>3. Если не возвращено сообщение об ошибке, то выводится сообщение об успешном изменении</li> </ol>	<ol style="list-style-type: none"> <li>1. Сервис получает идентификатор услуги и значения необходимых атрибутов</li> <li>2. Если валидация не прошла успешно, то возвращается сообщение об ошибке</li> <li>3. Сохраняет изменённые атрибуты</li> </ol>
Удаление услуги	Администратор	<ol style="list-style-type: none"> <li>1. Пользователь выбирает</li> </ol>	<ol style="list-style-type: none"> <li>1. Сервис получает идентификатор услуги</li> </ol>

		2. Если не возвращено сообщение об ошибке, то выводится сообщение об успешном удалении	2. Если на данную услугу есть записи, то возвращается сообщение об ошибке при этом услуга не удаляется 3. Удаление услуги при отсутствии ошибок
--	--	--	--

### 3. Сущность «Запись на услугу»

#### 3.1 Глоссарий атрибутов

Атрибут	Условия	Обязательность для заполнения
Услуга	-	Да
Пользователь	-	Да
Дата и время оказания	Формат даты	Да
Признак активности	Оказана/Активна/Отменена	Да

#### 3.2 Функции

Наименование	Роль	Пользовательский сценарий	Функциональный сценарий
Просмотр своих записей на услуги	Зарегистрированный Пользователь	1. Пользователь переходит на страницу просмотра списка своих записей на услуги 2. Пользователь имеет возможность применить следующие фильтры: Признак активности, Запись на конкретную дату	1. Сервис получает идентификатор пользователя 2. Сервис выводит список записей пользователя на услуги с учетом выбранных фильтров и сортировки по атрибуту «Дата и время оказания»
Просмотр записей на услуги	Администратор	1. Пользователь переходит на страницу просмотра списка записей на услуги 2. Пользователь имеет возможность применить следующие фильтры: Признак активности, Запись на конкретную дату	1. Сервис выводит список записей на услуги с учетом выбранных фильтров и сортировки по атрибуту «Дата и время оказания»
Добавление новой записи	Зарегистрированный Пользователь	1. Выбирает необходимую услугу	1. Сервис получает идентификатор услуги, идентификатор

		<ol style="list-style-type: none"> <li>2. Пользователь вводит желаемую дату и время</li> <li>3. Если не возвращено сообщение об ошибке, то происходит перенаправление на страницу просмотра своих записей на услуги</li> </ol>	<p>пользователя и значение необходимых атрибутов</p> <ol style="list-style-type: none"> <li>2. Если нет свободных мест, либо выбранное время занято, либо выбрана прошедшая дата, то возвращается сообщение об ошибке</li> <li>3. Происходит запись на услугу</li> </ol>
Изменение даты и времени оказания услуги	Зарегистрированный Пользователь	<ol style="list-style-type: none"> <li>1. Пользователь выбирает свою запись на услугу</li> <li>2. Пользователь вводит изменённые значения даты и времени</li> <li>3. Если не возвращено сообщение об ошибке, то выводится сообщение об успешном изменении</li> </ol>	<ol style="list-style-type: none"> <li>1. Сервис получает идентификатор записи на услугу и значения даты и времени</li> <li>2. Если валидация не прошла успешно, либо нет свободных мест, либо выбранное время занято, либо выбрана прошедшая дата, либо редактирование проводится позже чем за сутки (24 часа), то возвращается сообщение об ошибке</li> <li>3. Сохраняет изменённые атрибуты</li> </ol>
Отмена своей записи на услугу	Зарегистрированный Пользователь	<ol style="list-style-type: none"> <li>1. Пользователь выбирает свою запись на услугу</li> <li>2. Если не возвращено сообщение об ошибке, выводится сообщение об успешной отмене</li> </ol>	<ol style="list-style-type: none"> <li>1. Сервис получает идентификатор записи на услугу</li> <li>2. Если отмена записи производится позже чем за сутки (24 часа), то возвращается сообщение об ошибке</li> </ol>



			3. Изменяется атрибут «Признак активности»
Редактирование записи на услугу	Администратор	<ol style="list-style-type: none"> <li>1. Пользователь выбирает запись на услугу</li> <li>2. Пользователь вводит изменённые значения необходимых атрибутов</li> <li>3. Если не возвращено сообщение об ошибке, то выводится сообщение об успешном изменении</li> </ol>	<ol style="list-style-type: none"> <li>1. Сервис получает идентификатор записи на услугу и значения необходимых атрибутов</li> <li>2. Если валидация не прошла успешно, либо нет свободных мест, либо выбранное время занято, либо выбрана прошедшая дата, то возвращается сообщение об ошибке.</li> <li>3. Сохраняет изменённые атрибуты</li> </ol>
Отмена записи на услугу	Администратор	<ol style="list-style-type: none"> <li>1. Пользователь выбирает запись на услугу</li> <li>2. Выводится сообщение об успешной отмене</li> </ol>	<ol style="list-style-type: none"> <li>1. Сервис получает идентификатор записи на услугу</li> <li>2. Изменяется атрибут «Признак активности»</li> </ol>

## Модуль 4 (3 часа)

*Вам необходимо улучшить работу сервиса путем изменения взаимодействия пользователя с интерфейсом.*

Для этого необходимо реализовать следующий функционал:

- Пагинация страницы с услугами
- Поиск по услугам с автодополнением (подсказки при наборе текста) слов в поиске(ajax)
- Всплывающие уведомления о предстоящих услугах на следующий день.
- Импорт данных об услугах из XML файла - предоставляется файл, необходимо написать страницу для его импорта в БД. При импорте данных об услугах в качестве логотипа используется графическая «заглушка» (файл default.jpg)
- Улучшение дизайна, учитывая целевую аудиторию. Создание логотипа.
- Реализовать функционал формирования и выгрузки отчетности в формате XML (список оказанных услуг, список оказанных услуг за определенный промежуток времени и список отмененных записей на услугу)
- Возможность изменения порядка вывода списка услуг с помощью Drag-and-Drop

Разрешенные PHP-фреймворки: Yii2 (basic), Laravel 5

## **Задание третьего дня (Wordpress)**

*Вам необходимо разработать блог «Дневник космического автосервиса». Этот блог служит для привлечения пользователей к использованию разработанного ранее сервиса.*

Для этого необходимо установить CMS (Wordpress), настроить её и доработать согласно заданию. Вам предоставляется архив с CMS, а также набор плагинов, шаблонов, текстов и изображений, расположенных в папке media.

## **Модуль 5 (3 часа)**

В данном модуле необходимо:

- произвести установку CMS
- изменить путь к панели администрирования - должна быть доступна по адресу <сайт>/admin (логин admin, пароль admin)
- настроить её согласно предоставленным данным о компании
- заполнить предоставленной информацией (разрешается дописывать информацию для создания целостности восприятия сайта).
- установить плагины: галерея, обратная связь, плагины для продвижения ресурса
- установить шаблон - один из предоставленных. Разрешается изменять шаблон, добавлять (удалять) картинки и блоки, использовать анимацию для создания лучшего восприятия сайта
- в футере сайта представить краткую информацию об организации
- организовать доступ к меню сайта при прокрутке страницы