

Поправителен изпит по Теория на програмирането - практика

(16-22 юли 2018)

Задание

Вариант 1

Редуцирано условие за задача за среден 3 при реализация за 35 минути. Позволява се използване на интернет. Позволява се предварително подготвени подобни задачи и техните условия да се достъпват. Такива има поне 15 дадени през учебния срок

1. Create Task on url /tasks/new with the following field

- name
- description
- solution_required - true/false value if a solution should be provided for the task to be marked as completed. A check box in the form

Implement the TasksController to have a show, update, create, edit, destroy, index methods that follow the rails convention for behaviour of these methods

2. Create TaskSolution /task_solutions/new with

- picture solution - field for providing a picture solution with an upload button in the form
- status - enum of values { :not_started, :started, :completed }. A drop down in the form
- confirmed - true/false value if the task is confirmed

Implement the TaskSolutionsController to have a show, update, create, edit, destroy, index methods that follow the rails convention for behaviour of these methods

3. One task could have many task solutions. One task solution is for a specific task

4. Validate that the number of task solutions created for a given task that have a blank text solution is not more than 3

5. Validate that no more than 3 tasks requiring a solution could be created

6. Validate that for a task to be marked as completed when a solution is required there should be a picture solution provided and that after a task is marked as confirmed the picture and status could not be changed.

7. Show a list of task completions and for which task they are

- this is available at /#{my_index}tasks/ where my_index is your class_number_firstname_lastname

- show a table with the first column the task name, the second column the specialty description, the third column whether a solution is required, the fourth column the text task solution, the fifth column the status of the solution and the sixth column whether the task is confirmed

Вариант 2

Редуцирано условие за задача за среден 3 при реализация за 35 минути. Позволява се използване на интернет. Позволява се предварително подготвени подобни задачи и техните условия да се достъпват. Такива има поне 15 дадени през учебния срок

1. Create Task on url /tasks/new with

- name
- description
- solution_required - true/false value if a solution should be provided for the task to be marked as completed

Implement the TasksController to have a show, update, create, edit, destroy, index methods that follow the rails convention for behaviour of these methods

2. Create TaskSolution /task_solutions/new with

- text solution - field for providing a text solution
- status - text. A text field in the form
- confirmed - integer value if the task is confirmed. 1 confirmed. 0 not confirmed. A check box in the form

Implement the TaskSolutionsController to have a show, update, create, edit, destroy, index methods that follow the rails convention for behaviour of these methods

3. One task could have many task solutions. One task solution is for a specific task

4. Validate that the number of task solutions provided for a task are less than or equal to 2

5. Validate that no more than 2 tasks with solution_required=false could be created for a given task

6. Validate that confirmed should be less than or equal to 1 or more than or equal to 0

7. Show a list of task completions and for which task they are

- this is available at /#{my_index}tasks/ where my_index is your class_number_firstname_lastname

- show a table with the first column the task name, the second column the specialty description, the third column whether a solution is required, the fourth column the text task solution, the fifth column the status of the solution and the sixth column whether the task is confirmed

Вариант 3

Пълно изискване за оценки от 3 от 6, достъпно публично поне една седмица преди изпита.

Редуцирано условие за задача за среден 3 при реализация за 35 минути. Позволява се използване на интернет. Позволява се предварително подготвени подобни задачи и техните условия да се достъпват. Такива има поне 15 дадени през учебния срок

Да се реализира web платформа за продажба на документи. Част от документите са налични публични и всеки един потребител може да има достъп до тях, част от документите изискват потребителят да е заплатил за достъпа си.

Платформата да поддържа

Следните функционалности за средна оценка 3

1. Регистрация на потребител – потребител се регистрира с потребител и парола, получава писмо за регистрацията си и потвърждава регистрацията си като избира и последва връзка в писмото. Така се удостоверява, че мейлът е негов.
2. На началната страница се показват последните три качени документа в платформата както и дали те изискват абонамент или не, тяхното име и тяхното кратко описание
3. Потребителят може да отиде на адрес /documents и да види всички документи в платформата, като те са разделени на страници от по 10. Поддържа се pagination
4. Администраторите в платформата имат право да отидат на /documents/new където се показва форма за създаване на нов документ. Посочват се име на документ, картинка, кратко описание, самият документ се качва, както и дали документът изисква плащане или не. Избира се бутон Create и документът се създава в платформата и обикновенните потребители вече могат да го достъпват
5. Всеки отделен документ може да се достъпи на url-то /documents/:id където id е идентификаторът на документа в базата.
6. Когато потребител достъпи адрес на документ в браузъра той може да има достъп до него или да няма достъп. Ако има достъп му се показват картинката на документа, името, описанието и връзка за свалянето на документа. Ако няма достъп му се показват картинката, името, описанието и съобщение, че той няма достъп и затова трябва да се абонира.
7. Потребител може да отиде на страницата /subscriptions където е показана информация за плана и да закупи нов абонамент за този план. Закупуването на абонамент се прави като потребителят отговори на въпроса колко е „2+2“. Ако потребителят правилно е отговорил на този въпрос се счита, че си е закупил абонамент. Целта е да няма връзка към външни услуги за плащания. Ние да си управляваме плащанията по възможно на-простия начин.
8. Един потребител може да закупи даден план само веднъж. Ако вече го е закупил трябва да му се покаже информация кога го е закупил.

9. Ако потребител има редовно закупен план трябва да може да достъпва всички документи, които изискват закупен план.
10. Приложението да се качи на сървър и да е достъпно през интернет. Примерно Amazon, Heroku, Azure

Следните допълнителни функционалности за оценка 4

1. Да се създадат 3 плана и човек да може да се абонира за повече от един план. Първият план да дава възможност на потребител да достъпва документите в рамките на следващите 1 минута. Вторият в рамките на следващите 2 минути. Третият в рамките на следващите 3 минути. План 1 се закупува с отговор на въпроса колко е 1+1, план 2 се закупува с отговор на въпроса колко е 2+2 и план 3 се закупува с отговор на въпроса колко е 3+3
2. За всеки един документ да се знае кои са авторите. Един документ може да има много автори. В страницата за документа се показват имената на авторите на документа и html връзка, която да води към адрес /authors/:id където id е идентификатора на автора
3. Като се посети адрес /authors/1 може да се видят всички документи публикувани от автор 1.

Следните допълнителни функционалности за оценка 5

1. За всеки един документа да може да се посочи до кой точно абонамент трябва да се е абонирач човека, за да може той да има достъп до него.

Следните допълнителни функционалности за оценка 6

1. Да се добавят категории достъпни на /categories. Всеки документ може да бъде в една или повече категории. Като се посети адресът /categories/1 да се покажат всички документи в тази категория.
2. Да се добави валидация, че един автор не може да има повече от 70% от документите си като платени.
3. Да се праща съобщение на авторите кои са документите които днес са били свалени от потребителите и колко пъти е бил свален всеки документ