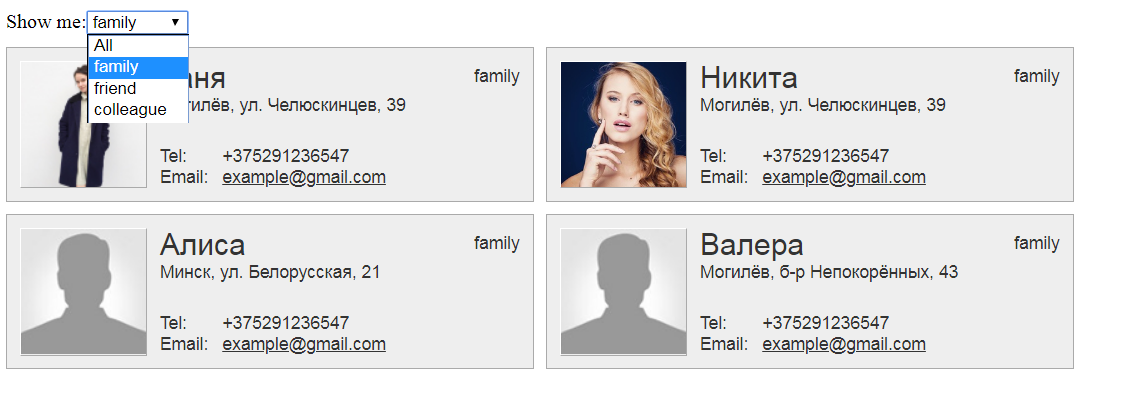
# Лабораторная работа №14. Менеджер контактов

# на Backbone.js

Разработать форму, добавление /отображение контактов\_ группировка по следующим категориям: друзья, подруги, коллеги, семья.

Типа такого:

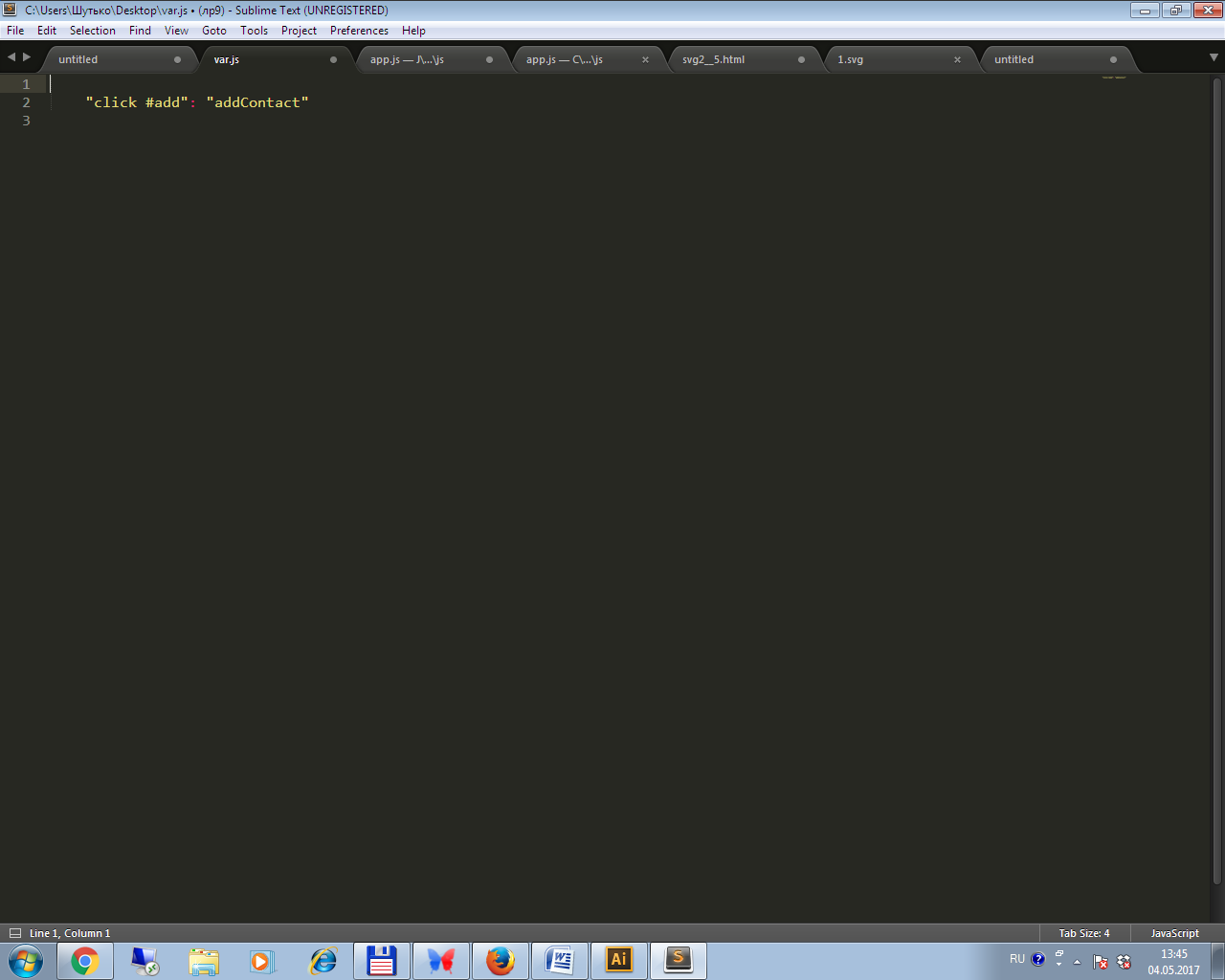


При выборе категории, отображаются отобранные контакты

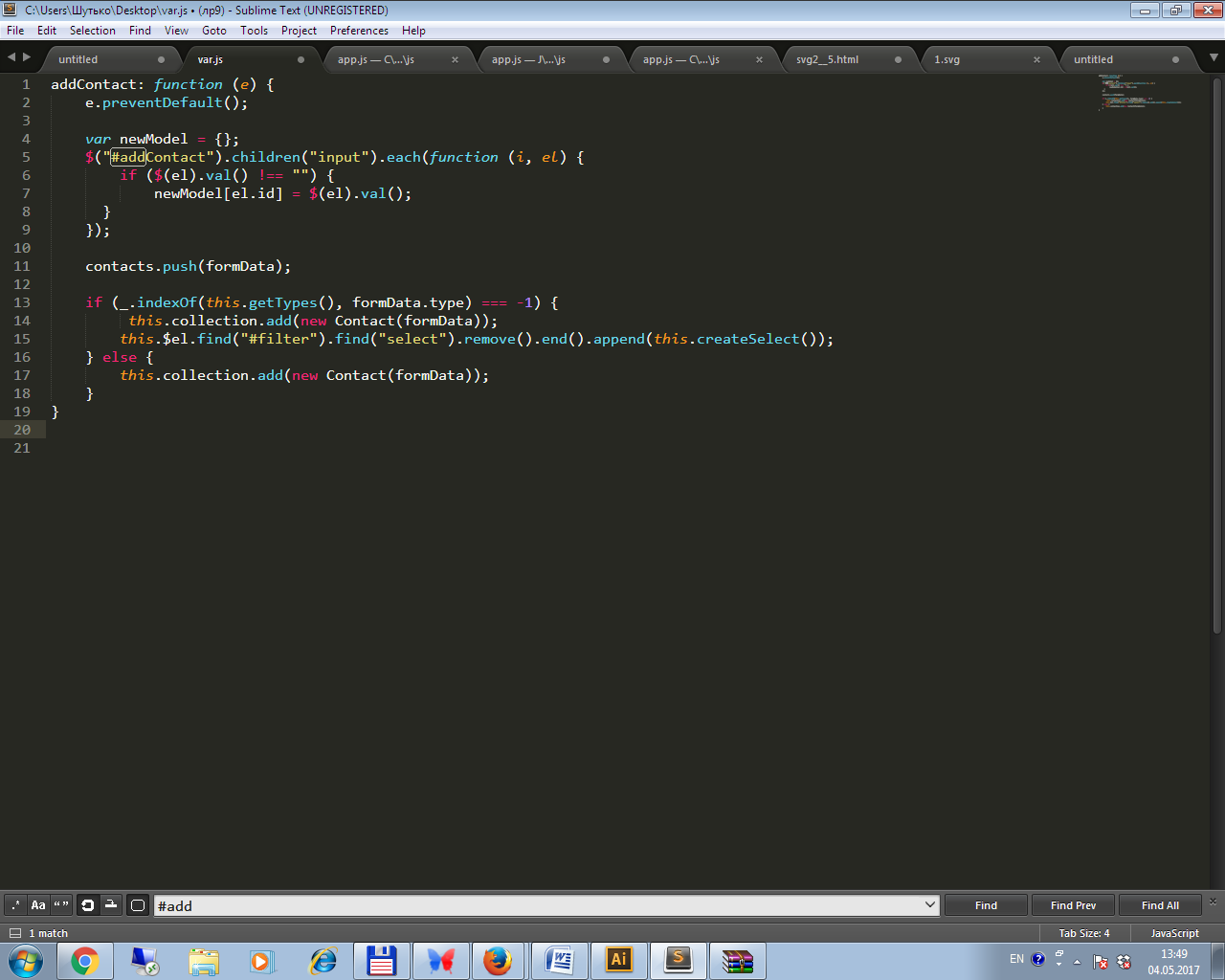
1.Добавить модели в коллекцию

Создайте форму с id = «addContact», которая должна содержать следующие области: фото, тип, имя, адрес, телефон, email, а также кнопку добавить. Необходимо следить за тем, чтобы значения атрибутов элементов <input> совпадали с теми полями, что вы используете в модели.

Затем необходимо добавить обработчик события в главное представление для того, чтобы можно было собрать данные при отправке формы; добавьте следующий код после всех существующих элементов типа ключ:значение в объекте событий:

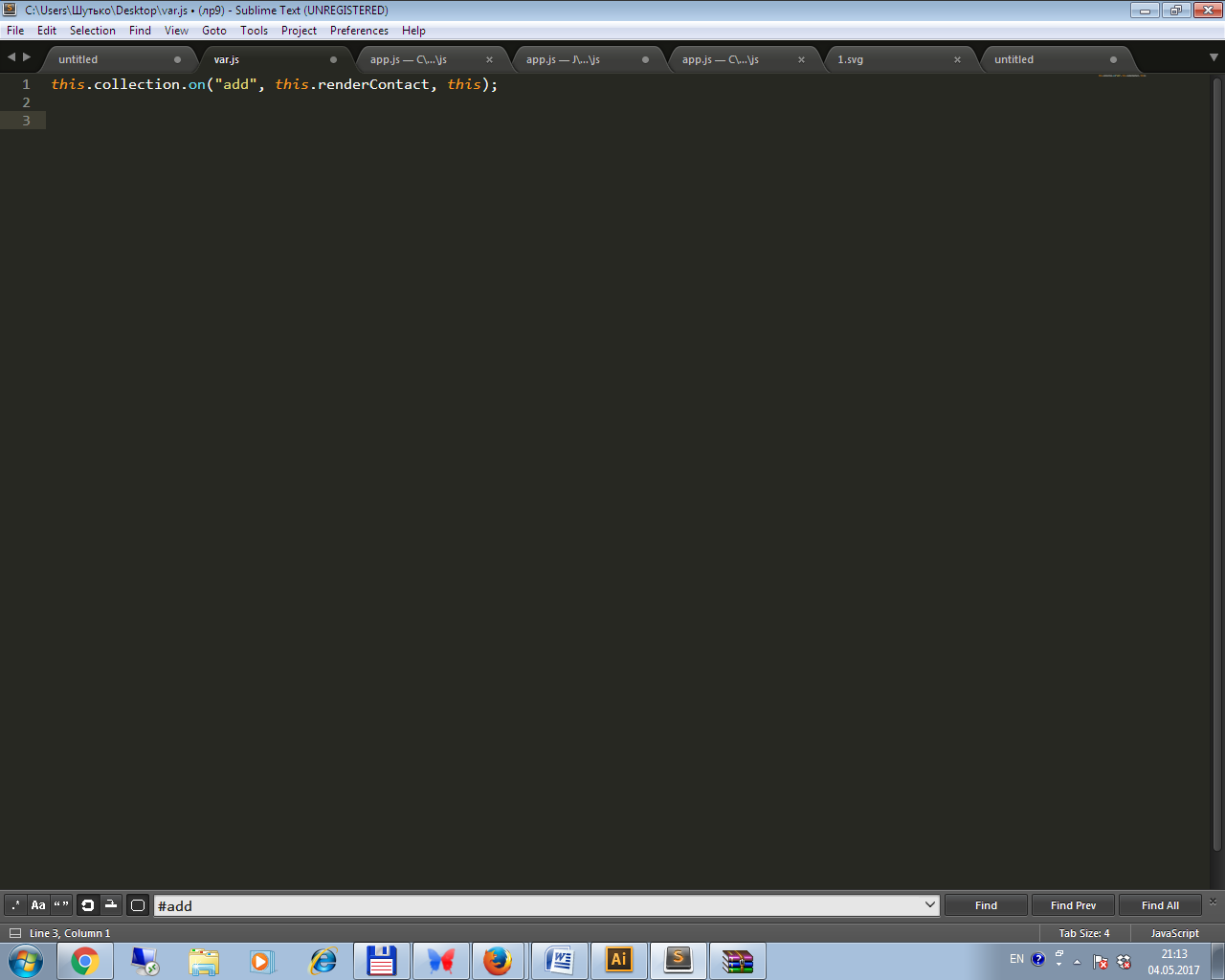


Будем отслеживать событие клика по элементу с id равным add (кнопка в форме). Обрабатывать клик мы будем в методе addContact. Добавьте следующий код после метода thefilterByType:



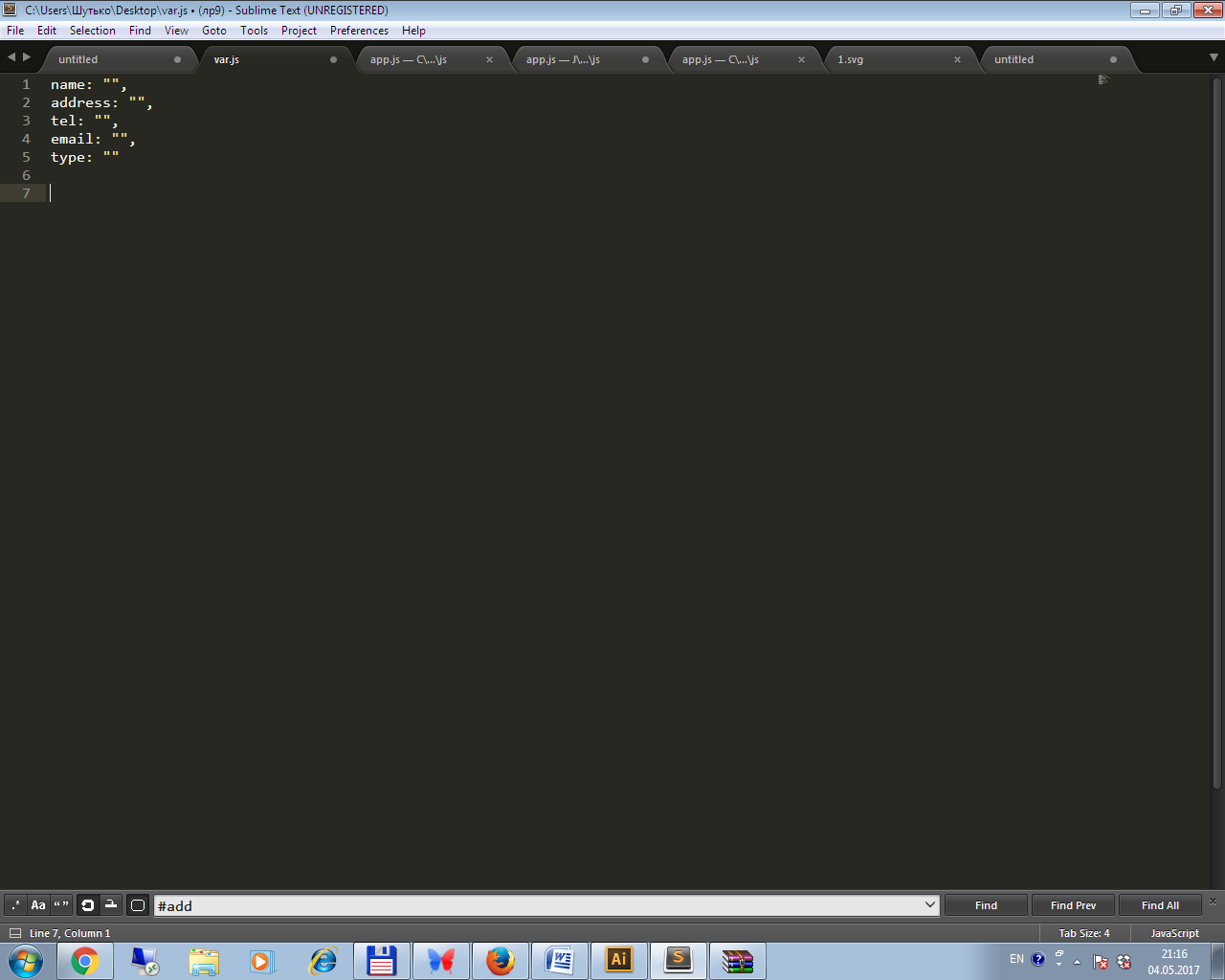
2. Отобразить новый контакт на странице

Теперь необходимо отобразить новую модель на странице. Для этого потребуется ещё один обработчик события, прикрепим его к событию add. Добавьте следующий фрагмент в метод initialize() коллекции:



Метод on() используется для отслеживания добавления нового контакта и вызова уже существующего метода для формирования html кода, который будет отображён на странице.

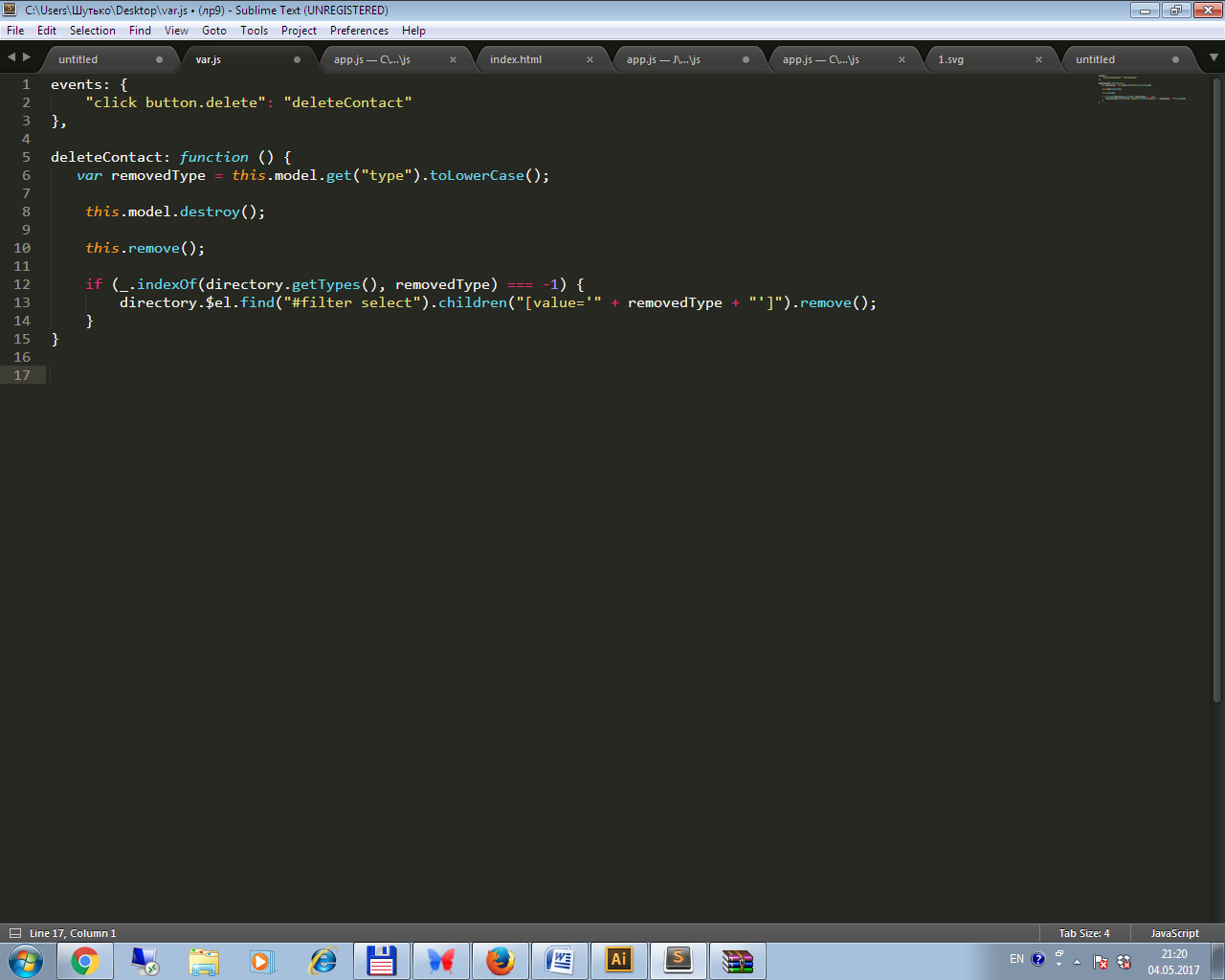
Если все поля формы addContact будут пустыми, то в дальнейшем могут возникнуть проблемы с моделью. Одно из решений - это выставление значений по умолчанию, в случае если данные не введены. Если это невозможно сделать, то в качестве значений можно передать пустые строки. Обновите объект класса Contact для назначения значений полей по умолчанию:



3. Удалить модель из коллекции

Создайте для каждого контакта свою кнопку удаления.

Процесс удаления модели может быть добавлен в класс представления, который является по сути каждым отдельным контактом. Итак, нам нужно добавить событие и обработчик события “клика по кнопке” удаления; добавьте следующий код в конец класса ContactView:



Мы используем объект событий для того чтобы назначить перечисленное событие. На этот раз мы слушаем событие клика по элементу <button> для удаления контакта. Обработчик данного события метод deleteContact.

Для начала мы сохраняем тип контакта, который хотим удалить. Преобразуем данное значение к нижнему регистру, чтобы избежать непредвиденных ситуаций.

Затем вызываем метод destroy() от объекта модели. Мы так же удаляем HTML код, отображающий контакт на странице с помощью jQuery метода remove(), а так же очищаем все события и обработчики событий, ассоциированные с данным элементом.

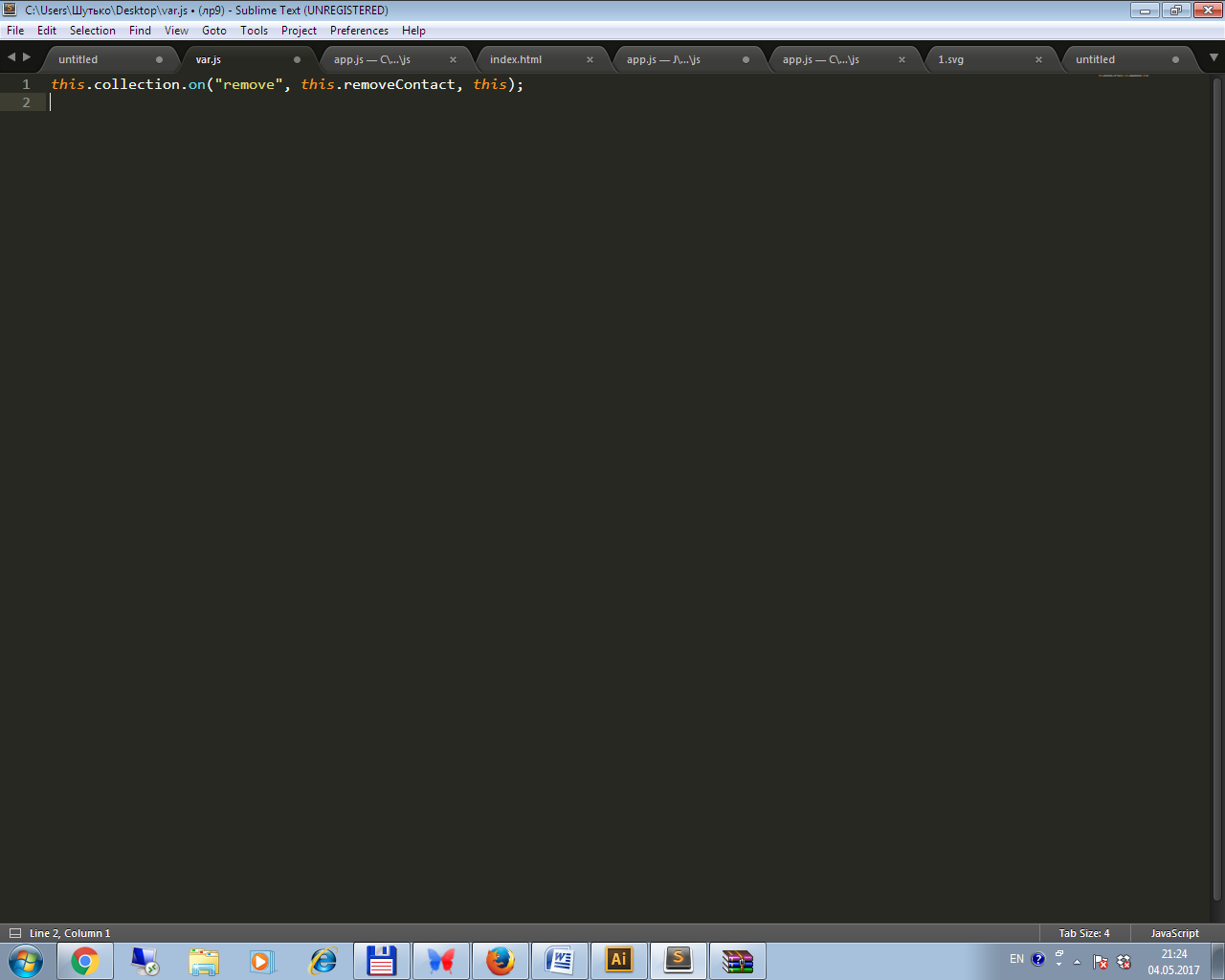
Напоследок, мы проверяем тип всех моделей в коллекции для того чтобы знать оставить ли тип, удалённого контакта в элементе <select>. Если никакая модель не относится к данному типу, то преобразуем элемент <select> и отображаем его новое содержимое на странице.

Извлекаем элемент <option> с значением в атрибуте value, совпадающее с значением переменной removedType, которое мы сформировали раньше.

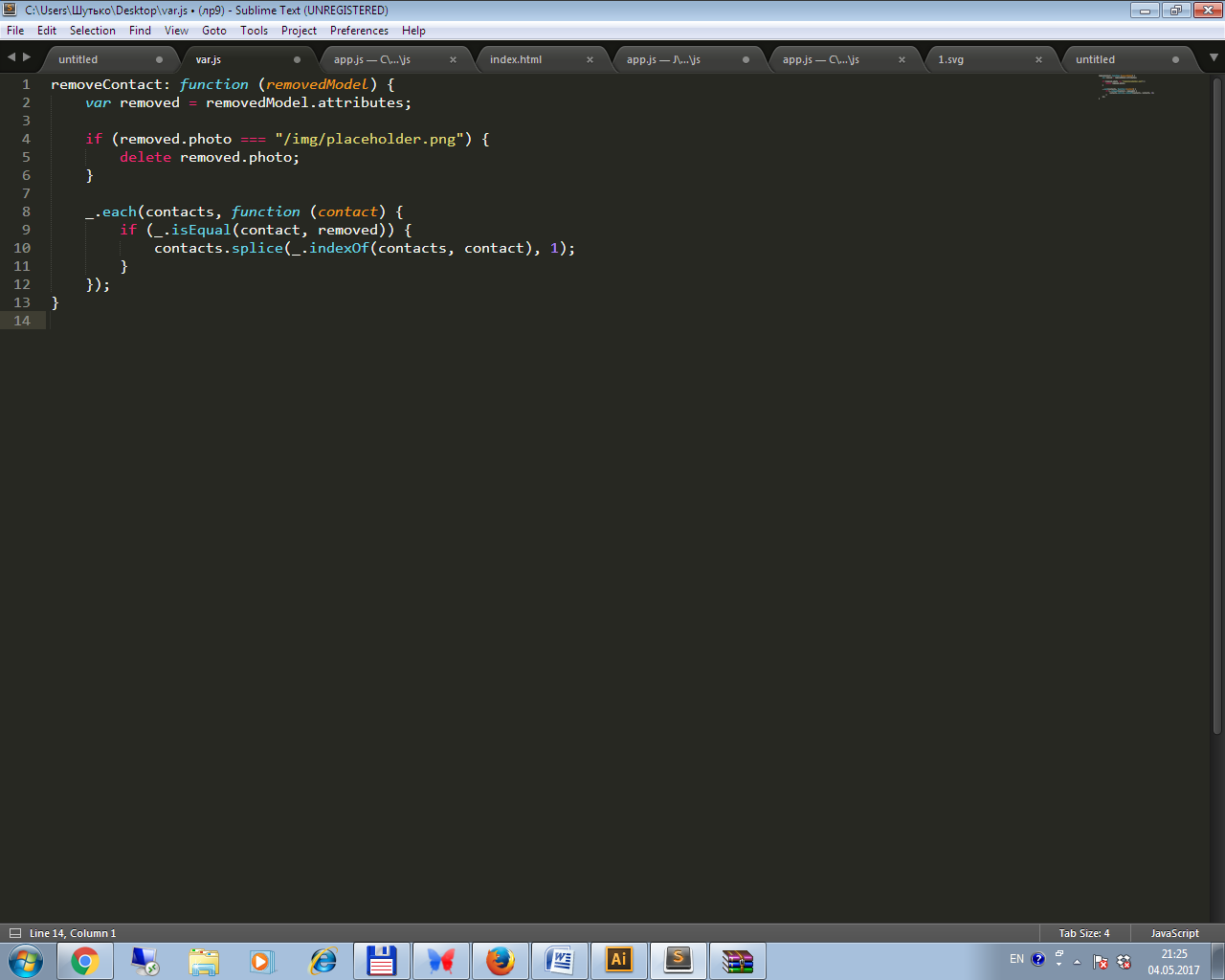
4. Удалить данные из модели

Помимо удаления контакта из коллекции и представления, необходимо удалить его из массива. Если этого не сделать, то модель может возникнуть позже, при фильтрации.

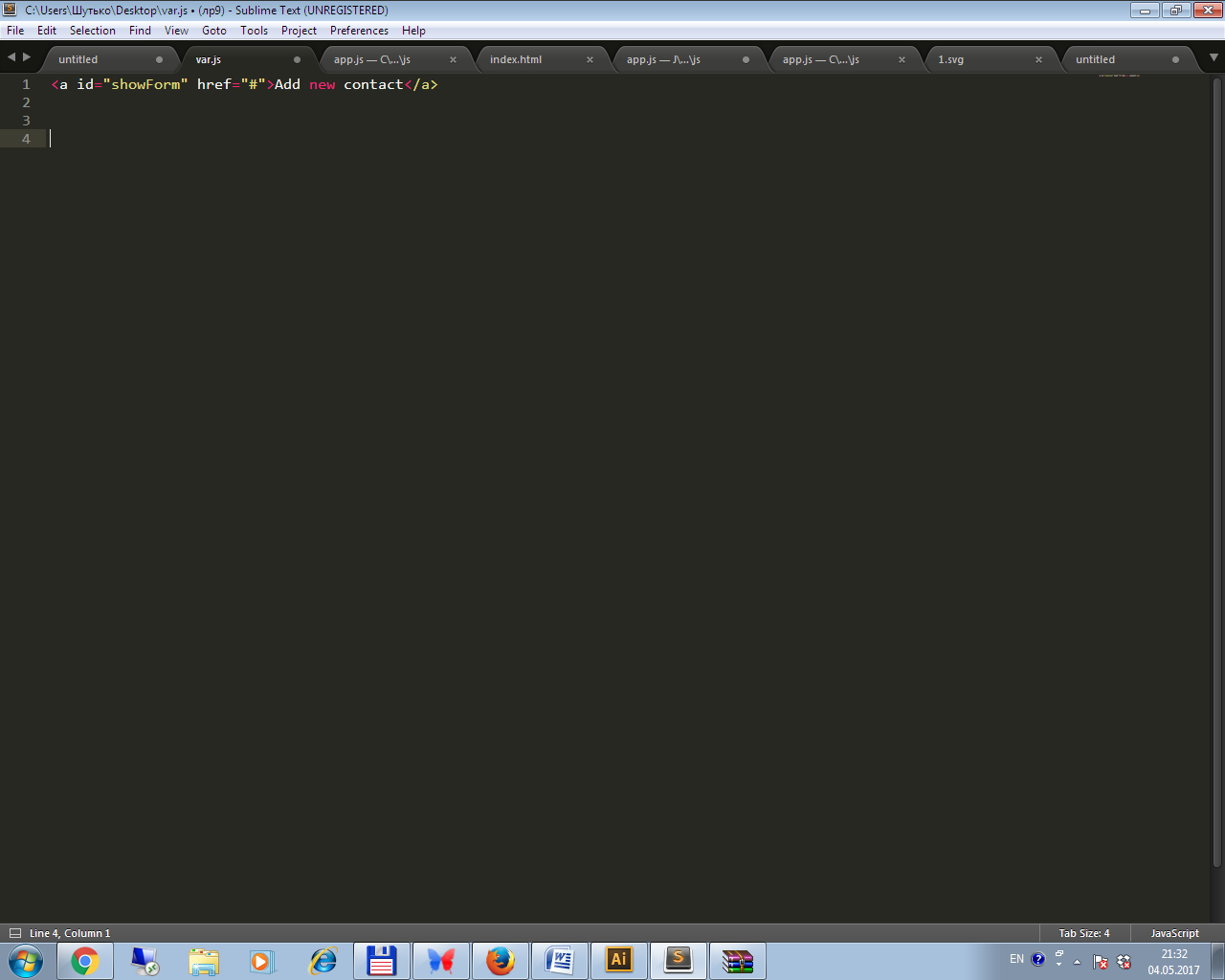
Функционал удаления данных из оригинального массива можно расположить в главном представлении; к коллекции мы прикрепим событие remove и назначим обработчик, для того чтобы мы могли быстро определить когда нужно удалить данные из массива. Добавьте следующий код сразу после существующих привязок событий к модели:



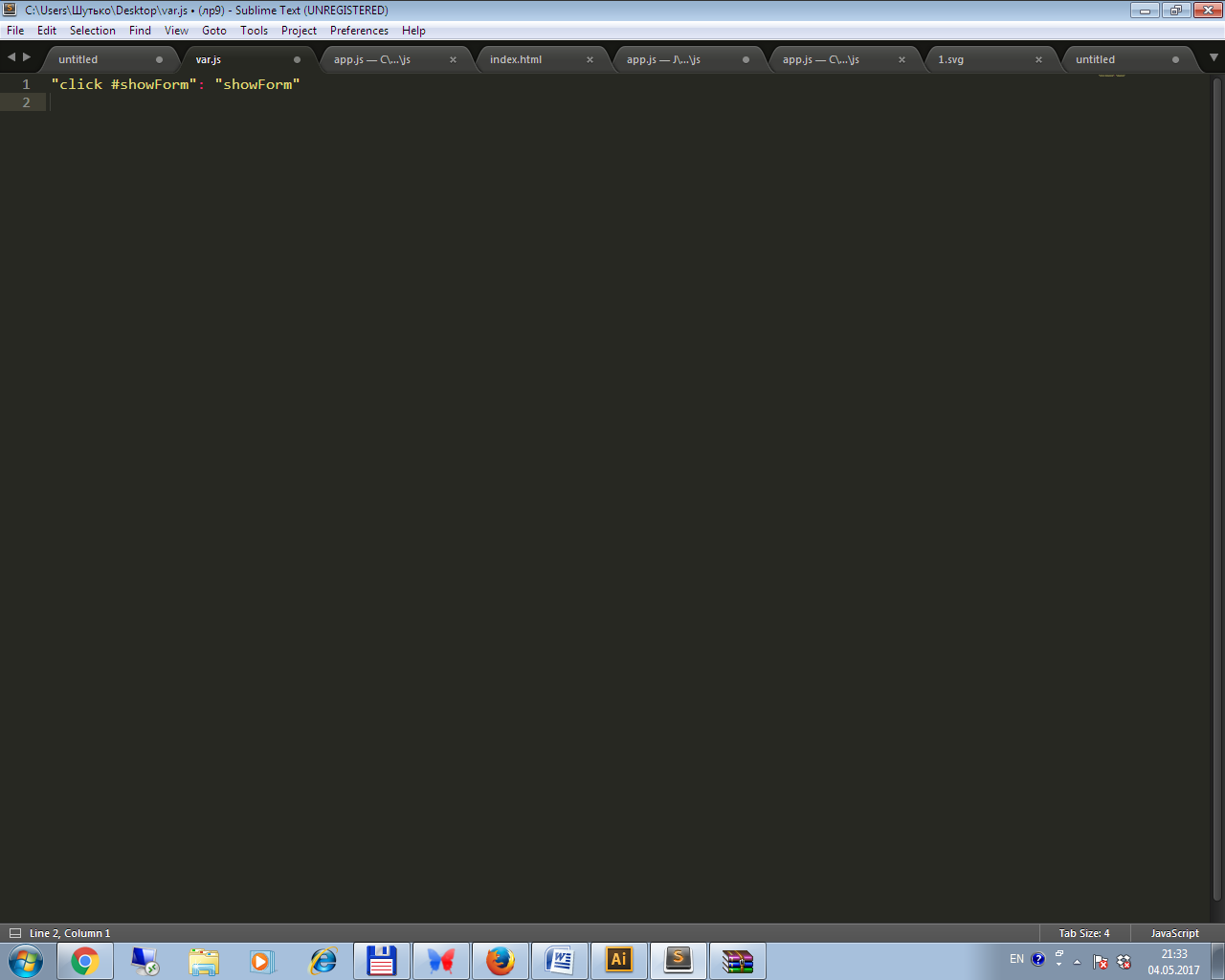
Далее создаём метод removeContact(); добавьте его после метода addContact():

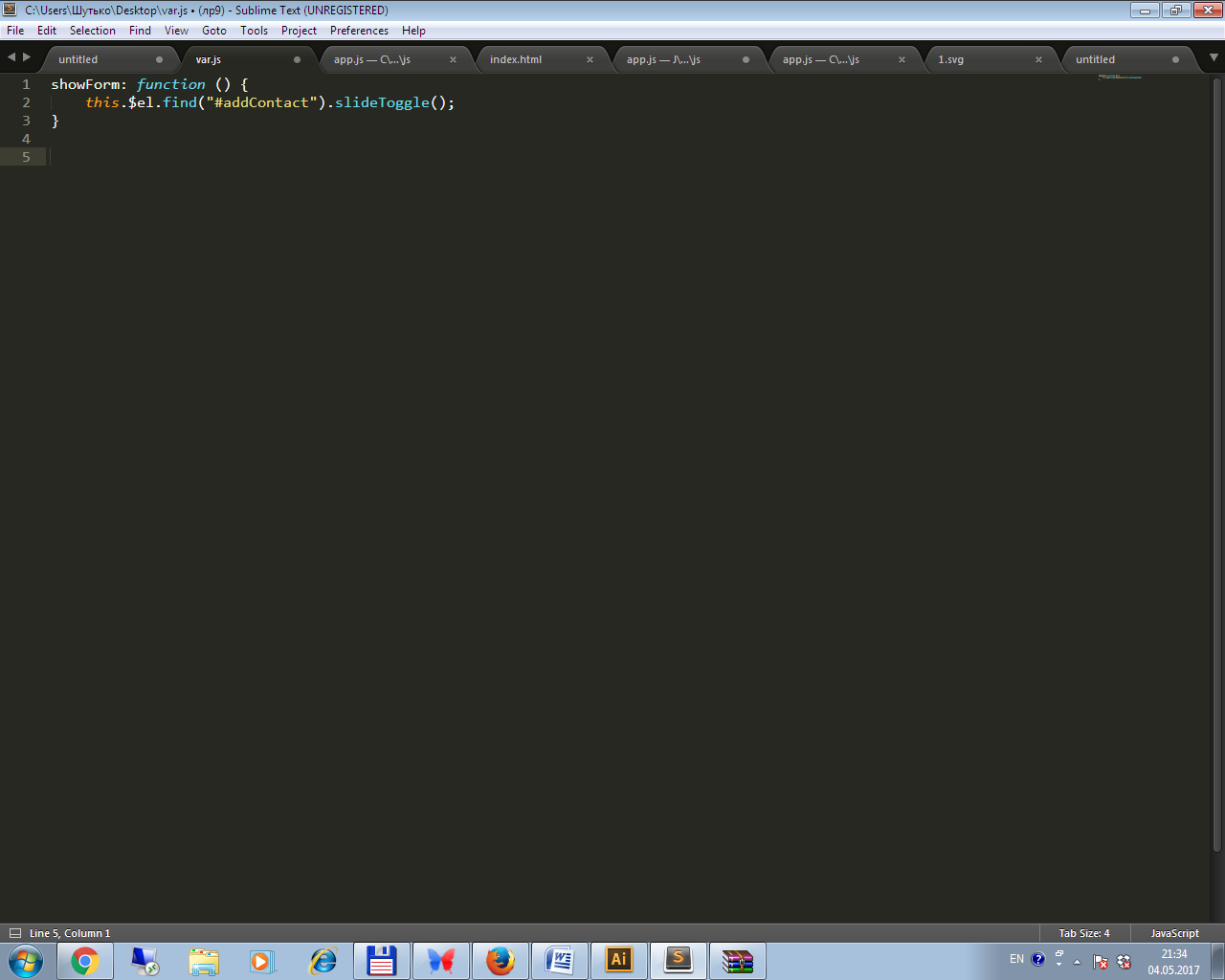


Напоследок, спрячем форму и будем отображать её при клике на специальную кнопку. Добавьте следующий код к элементу <header>:



Для начала нужно скрыть форму и отображать при клике на специальный элемент. Добавим обработчик события к классу DirectoryView:

Метод showForm():



Дополнительно:  
Приветствуется своя форма для менеджера контактов.

|  |  |
| --- | --- |
|  |  |