

SOME HINTS FOR PROJECT MILESTONE 1

Checkbox

Option 1

```
<input name="box" type="checkbox" value="dummy">
```

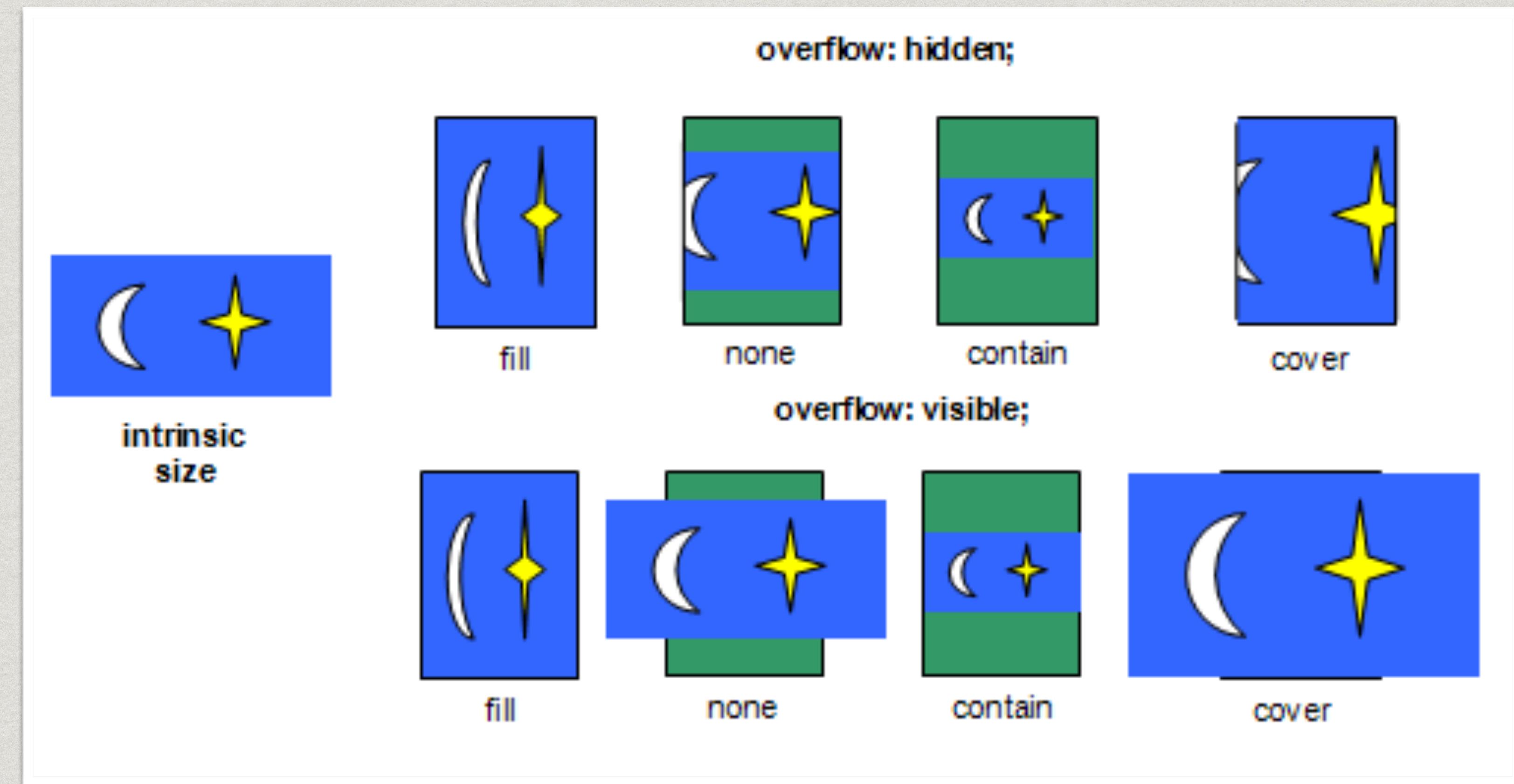
- * When this checkbox is clicked, this form element, **together with its value**, is submitted to the server.
- * If the box is NOT clicked, this form element won't be passed.



```
<input name="box" type="checkbox" value="dummy" checked>
```

- * The box would be checked if the **checked** attribute is provided in the opening tag.

CSS Object-fit



<https://www.w3.org/TR/2011/WD-css3-images-20110217/#object-fit0>

QuerySelector

- * The Document method `querySelector()` returns the first Element within the document **that matches the specified selector, or group of selectors.**
- * If no matches are found, null is returned.
- * For example:

```
document.querySelector('option[value="Elements"]')
```

Selector

- * Read: <https://api.jquery.com/category/selectors/>

Element Selector (“element”)

Selects all elements with the given tag name.

Class Selector (“.class”)

Selects all elements with the given class.

ID Selector (“#id”)

Selects a single element with the given id attribute.

Attribute Equals Selector [name="value"]

Selects elements that have the specified attribute with a value exactly equal to a certain value.

Date Object in JavaScript

- * The **Date** object is used to work with dates and times.
- * Date objects are created with **new Date()**.
- * There are four ways of instantiating a date:

```
var d = new Date();
var d = new Date(milliseconds);
var d = new Date(dateString);
var d = new Date(year, month, day, hours, minutes, seconds, milliseconds);
```

- * To language sensitive **representation of the date portion of this date**, we may use

```
new Date().toLocaleDateString();
```

Date and Range Checking

- * Under Sails, a **date** could be saved as type **“ref”**, with a special **columnType**.
- * Indicates the type of **physical-level column** data type to use for an attribute when Sails creates the database table.
- * Finally, here's an example of **range check** in Controller actions:

```
Model.find({ age: { '>=': 21, '<=': 25 } })
```

```
module.exports = {  
  attributes: {  
    attr1: {  
      type: 'ref',  
      columnType: 'datetime'  
    },  
    attr2: {  
      type: 'ref',  
      columnType: 'datetime'  
    }  
  }  
};
```

MILESTONE 2

- * Your milestone 1 is well received, thank you.
- * Now, you can **continue to polish it**, as the **completeness of Milestone 1 also plays a part in Milestone 2**.

Milestone 2

- * Develop a new api (controller/model) to handle the **user information**.
- * Provide login and logout
 - * Store the **username and id in the session**, you will need them in multiple actions.
- * Setup a **many-many association** with the first model in your system.

- * There are **three types of users** in our system

System admins

- * Able to add, update and delete the real estate properties.
- * Able to **see the occupants of each real estate property.**

Clients

- * Clients can **co-rent an apartment, subject to availability.**
- * Clients can check his/her own **rental status.**

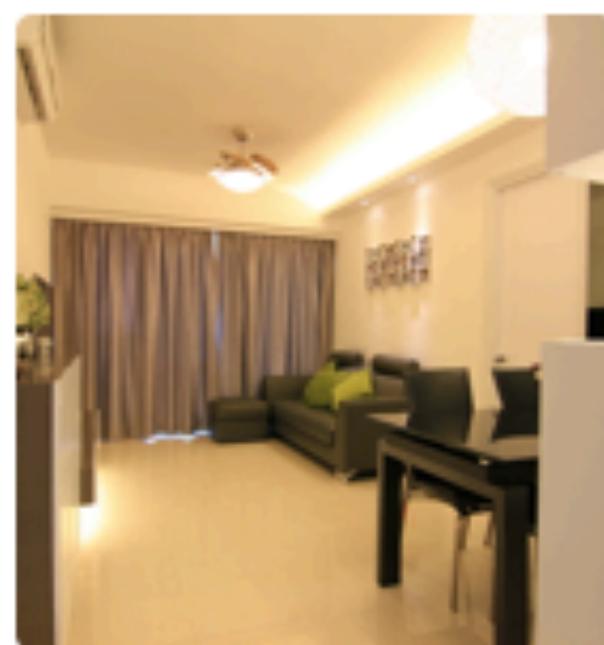
Everyone

- * Can browse and search for real estate properties.

Real Estate Rental System

Real Estate Rental Management System is a web-based system to facilitate flatshare and finding roommates.

平絕同區



Estate: Festival City

Gross Area: 700 sq feet

Rent: \$15000

Real Estate Rental System

Real Estate Rental Management System is a web-based system to facilitate flatshare and finding roommates.

show detail
page now

		Festival City	Occupants	Edit	Delete
1	酒店式靚裝，有泳池會所	Tin Ma Court	Occupants	Edit	Delete
2	2房實用，交通方便	City One Shatin	Occupants	Edit	Delete
3	沙田第一城 套3房剛翻新	Festival City	Occupants	Edit	Delete
4	平絕同區		Occupants	Edit	Delete

Book and Cancel

- * Add **co-rent button** in detail page.
- * Make sure **there's enough room from the properties.**
- * The button should be **replaced by a Move-out button** if the client has already rented the apartment.
- * Adopt the uses of **Ajax** and **restful API**.

- * Input validation
 - * Both **client-side and server-side** (model and controller action)
 - * Make sure input is valid to avoid system crashes.
- * **Disable the inappropriate API endpoints**
 - * GET only to retrieve data, not change server-side data.

- * Commit your work to GitHub **regularly**.
 - * I need to see your **development progress**.
 - * Use **better commit messages**.
- * Start earlier.
- * Due Date: **Nov 11, 20:00**.