

Release Notes: happycows, f24-12

The Happy Cows app has gained new features, has been cleared of many bugs both in terms of UI appearance and feature functionality, and has been bolstered with developer-friendly backend features that will make further development easy. Announcements now actually show up on the Play page based on their start and end date and admins can easily create new ones on the frontend. The Play page looks a lot more professional now with cleaner descriptions and table appearances and functional sorting and page scrolling. Backend operations for course rosters and Announcements have been added/refactored, so further development on these features will have a lot of support and almost no more backend work should have to be done.

The figure shows two side-by-side screenshots of a game interface. The left screenshot, titled 'Shown Button', shows a 'Show Announcements' button. The right screenshot, titled 'Hidden Button', shows a 'Show Announcements' button. Both screenshots display a welcome message for 'Yungong' and a table of announcements.

Bug fixes and UI/UX Improvements

- The “Mattanjah de Vries” link on the footer is replaced with the actual Professor About Page

The Link

The Page

- Fixed capitalization for “Buy” and “Sell” buttons, “cows” in “Buy Cows” and “Sell Cows”, and “Day” in “Today is Day...”.

Buying

Selling

Play Page

- Made the explanation of buying and selling cows neater.

Manage Cows

Market Cow Price: \$100

Number of Cows: 0

Current Milk Price: \$1

Buy cows

Sell cows

Note:

- Buying cows buys at the current cow price.
- Selling cows sells at the current cow price multiplied by the average health of cows (as a percentage)

- Previously the “Previous” and “Next” buttons in the Profits column persisted no matter what page you were on, now they dynamically appear based on page number. If you are not on the last page, the “Next” button will appear. If you are not on the first page, the “Previous” button will appear. If there is only one page, no buttons appear.

Profits

You will earn profits from milking your cows everyday at 4am.

Page: 1

Profit	Date	Health	Cows
\$0.00	2024-11-26	100.00%	0
\$0.00	2024-11-26	100.00%	0

Profits

You will earn profits from milking your cows everyday at 4am.

Page: 10

Previous

Next

Profit	Date	Health	Cows
\$0.00	2024-11-22	100.00%	0
\$0.00	2024-11-22	100.00%	0
\$0.00	2024-11-22	100.00%	0
\$0.00	2024-11-22	100.00%	0
\$0.00	2024-11-22	100.00%	0
\$0.00	2024-11-22	100.00%	0

Profits

You will earn profits from milking your cows everyday at 4am.

Page: 34

Previous

Profit	Date	Health	Cows
\$0.00	2024-11-25	100.00%	0
\$0.00	2024-11-25	100.00%	0
\$0.00	2024-11-25	100.00%	0
\$0.00	2024-11-25	100.00%	0
\$0.00	2024-11-25	100.00%	0
\$0.00	2024-11-25	100.00%	0

- The leaderboard now properly sorts by wealth. Before, it incorrectly ranked users with wealth less than \$1000.

Before

Farmer	Total Wealth	Cows Owned	Cow Health
Jonathan Zhang	\$199.00	99	100
Samuel Zhu	\$100.00	100	100
Aneesh Garg	\$10,000.00	0	100

After

Farmer	Total Wealth	Cows Owned	Cow Health
Yixuan Wong	\$10,000.00	0	100
Phill Conrad	\$9,904.00	1	100
Javin Zipkin	\$500.00	95	100

For Developers

OurTable.js Testing (OurTable.test.js file modified)

Modified OurTable.test.js so developers can test OurTable.js for line/branch coverage and mutations at a much faster rate. This will allow developers to test modifications made to the Leaderboard (OurTable.js) with much higher level efficiency. The runtime for mutation coverage sped up by about 9 minutes (see below).

Higher efficiency for testing as well (29 tests per mutant vs. 1.20 tests per mutant while still maintaining 100% mutation coverage).

Before

```
Ran 1.20 tests per mutant on average.
-----|-----|-----|-----|-----|-----|-----|
File      | % score | # killed | # timeout | # survived | # no cov | # errors |
-----|-----|-----|-----|-----|-----|-----|
All files | 100.00 | 2        | 77        | 0          | 0        | 0        |
OurTable.js | 100.00 | 2        | 77        | 0          | 0        | 0        |
-----|-----|-----|-----|-----|-----|-----|
09:46:30 (142996) INFO MutationTestReportHelper Final mutation score of 100.00 is greater than or equal to break threshold 100
09:46:30 (142996) INFO HtmlReporter Your report can be found at: file:///home/mayarosenbaum/cs156/proj-happycows-f24-12/frontend/reports/mutation/mutation.html
09:46:30 (142996) INFO MutationTestExecutor Done in 3 minutes 31 seconds.
```

After

```
Ran 29.09 tests per mutant on average.
-----|-----|-----|-----|-----|-----|-----|
File      | % score | # killed | # timeout | # survived | # no cov | # errors |
-----|-----|-----|-----|-----|-----|-----|
All files | 100.00 | 42       | 37        | 0          | 0        | 0        |
OurTable.js | 100.00 | 42       | 37        | 0          | 0        | 0        |
-----|-----|-----|-----|-----|-----|-----|
17:33:06 (96399) INFO MutationTestReportHelper Final mutation score of 100.00 is greater than or equal to break threshold 100
17:33:06 (96399) INFO HtmlReporter Your report can be found at: file:///home/mayarosenbaum/cs156/proj-happycows-f24-12/frontend/reports/mutation/mutation.html
17:33:06 (96399) INFO MutationTestExecutor Done in 12 minutes 46 seconds.
```

These modifications also yield 100% line/branch coverage (see below):

```
Test Suites: 68 passed, 68 total
Tests:       315 passed, 315 total
Snapshots:   0 total
Time:        14.26 s, estimated 22 s
Ran all test suites.
```

Announcements “Put” Operation Refactored

One of the broken CRUD operations for Announcements was Put. Swagger and testing revealed that it did not return proper error messages for different errors. It instead threw an error 400 “Bad Request” even in situations where this did not apply, such as a forbidden request error. The tests for Put were unrealistic and hid these issues. Both the operation and the tests have been remedied, and Put now appears the same for both Announcements and Commons. In a future

update, developers should find it significantly easier to implement and test the Edit page for announcements.

Creating Courses and being able to use post, get all and delete operations

The get-all, the post and delete operation are created for Courses. By creating these, an admin can choose to list all courses, add and delete any course.

courses

POST

/api/courses/post

Create a new course

GET

/api/courses/all

List all courses

DELETE

/api/courses

Delete a course

Creating Students and Being Able to Upload Student Roster

The get-all operation and the post-operation are created for Students. By creating these, an Administrator can upload a course roster to a specific course to add all students to a course.

Swagger Demonstration

Students

POST

/api/students/upload/egrades

Upload Students for Course in UCSD Egrades Format

Parameters

Name

Description

courseid

Integer (int32)

1

required

query

Request body

multipart/form-data

File

required

upload (binary)

Upload file

Execute

Clear

Course Roster

	Enrl Cd	Perm #	Grade	Final Units	Student Last	Student First Middle	Quarter	Course ID	Section	Meeting Time(s) / Location(s)	Email	ClassLevel	Major1	Major2	Date/Time	Pronoun
1	08235	A123456	4.0		GAUCHO	CHRIS FAKI	F23	CMPSC156	0100	T R 2:00- 3:15 SH 1431 W 5:00- 5:50 PHELP 3525 W 6:00- 6:50 PHELP 3525 W 7:00- 7:50 PHELP 3525	cgaucho@umail.ucsb.edu	SR	CMPSC		9/27/2023 9:39:25 AM	
2	08250	A987654	4.0		DEL PLAYA	LAUREN	F23	CMPSC156	0100	T R 2:00- 3:15 SH 1431 W 5:00- 5:50 PHELP 3525 W 6:00- 6:50 PHELP 3525 W 7:00- 7:50 PHELP 3525	ldeplaya@umail.ucsb.edu	SR	CMPSC		9/27/2023 9:39:25 AM	She (She/Her/Hers)
4	08243	I234567	4.0		TARDE	SABADO	F23	CMPSC156	0100	T R 2:00- 3:15 SH 1431 W 5:00- 5:50 PHELP 3525 W 6:00- 6:50 PHELP 3525 W 7:00- 7:50 PHELP 3525	sabadotarde@umail.ucsb.edu	SR	CMPSC		9/27/2023 9:39:25 AM	He (He/Him/His)