Capstone Project Ideas

note: most of these project ideas have not been tested, so it's unknown how well they'll work as capstone projects

1. Create an app to compute and visualize the places where a single-word changes will bring a passage of translated text closer in semantic similarity to the original meaning.
2. Create a web site that can manage code reviews. Show a diff view for two versions of a file, and allow comments on highlighted sections of text. Allow reviewers to propose changes that the original author can either accept or reject.
3. Personal Website Template. Create an interface by which a user can quickly specify, deploy and maintain a website with the following features: resume, image portfolio, web project demos, and list of publications. Focus on the user's workflow and keep boilerplate to a minimum.
4. Find an existing open source project and automate the maintenance workflows of it. Hook it up to Jira (or similar), Jenkins (or similar) and/or Travis CL. Automate bug closing when pull requests are merged that are designated as fixing a particular issue.
5. Create a load-balanced, autoscaled, autohealed system of VMs. We'd need to think of some workload for which load balancing would be needed. One possibility might be FEA simulations. Another might be distributed ray tracing. Someone from the art department might be able to make use of a render farm. This would be an introduction to load balancing. *(It needs to be somewhat unpredictable load, data source? 10s of VMs.)*
6. [one-pager here] Use bigquery to sort through large data sets and route the results into advanced information visualization. Bigquery has datasets for users to use. Suitable for text. Huge logs of events to visualize it.
7. Implement face recognition software such that could be used in face unlock, possibly using ML *(use their own photos?).* Possibly cluster photos from a large dataset to identify duplicates?
8. Create a cloud development environment based on CodeMirror or some other open-source browser-based editor. Use it to deploy a very simple android app from a chromebook.
9. Distributed rendering with deep compositing. Treat opacity as a per-pixel linear-piecewise function of depth for each channel (R,G,B). Render multiple scenes with transparency and combine them afterwards. Create a simple animation with one of the transparent parts moving over time, but keeping the rest fixed. (This was the bug report during production of Kung Fu Panda at DreamWorks that led to an academy award. Only suitable for someone really into graphics. For someone in that category, will be the right scale.)
10. Website to manage bookings for dog walkers and/or sitters. Similar to e-commerce, but the product is a service rather than a physical good, and there would need to be calendar coordination. Another possibility with almost identical requirements would be a reservation system for online language lessons.
11. Game authoring system like the recent Google doodle. Make it easy enough for a middle schooler to write a game for an elementary schooler.
12. a campus tour for the admissions office (using mapping functions on the phone to locate the person taking the tour)
13. combined app for attendance and quick polling that affect the results of a slide deck that is being used in a lecture
14. App for improving reading efficiency. It could be an extension or an app that makes web pages and articles quicker to read without editing it's content(no trimming or summarization).

Done before, but ok to do again because there's so many ways to do it.

- [ideas] Create an e-commerce website that can accept credit card payments *(making sth. secure enough, stripe, paypal, braintree, google wallet. Can be applied to social good, e.g., for local artists, animal shelters, recruit supplies and customers together)*. This has been done twice already, but it might be interesting to see this applied to services, rather than goods -- where the users have to deal with appointments.
- [one-pager here] Write a simple massively (or not so massively) multiplayer mobile game (possibly taking inspiration from netrek or pokemon go) *(shared state among many players hooked up with db, latency challenge, more broadly, can clone any existing video games. 50% of video game work is graphics)*
- Paper trading platform. Create an interface to virtually buy and sell stocks to practice strategies. Competition optional.
- Back test automatic stock trading strategies (based on ML?).
- Write a 2d web game along the lines of agar.io (a ton of people have made such games, but see if you can come up with a concept that's new).
- Write a simple mobile game. Get approved for ads, monetize it, and earn at least $1.
- Use TensorFlow to analyze resumes. Given plain text extracted by OCR, guess as much personal info about the person as you can (e.g. major, GPA).

Done before, and no good way that I can think of to extend

- CRM for off-market real estate leads. Write a mobile app to track addresses as they progress from lead to contact to offer to listing or sale. Assist investors in their "driving for dollars" activities.
- Implement a fantasy sports league. Players choose athletes to put on their "team", which then get points based on how those athletes do in real life.
- Web-based Property Management Software Suite. Create a website that can track maintenance requests from tenants and other tasks associated with property management. Implement a subset of the features in commercial products like Buildium or AppFolio
- social network for mutual accountability on exercise with a workout partner
- Write a bot that can deliberately waste peoples' time. Pass for human as long as you can get away with it. Bonus points if you can hook it up to voice synthesis and record some telemarketer calls.
- [ideas][one-pager here] Create a collaborative drawing app that can be accessed from web and mobile (a simple jamboard).
- Speaker preparation. When preparing for an important public speaking event, it is useful to practice quite a few times. Using text to speech, track deviations from a script, pace relative to previous runs (like those ghost joggers on fancy exercise machines that track your previous workouts). There might even be some way of giving a score to the fluidity of the speech. This could also be useful for students attempting to memorize a body of text (e.g. declaration of independence or preamble to the constitution)
- Write a cross-platform flutter app connected to firebase for managing check out/in of some communal physical items (e.g. board games, books, tools, etc...)
- Virtual graffiti. Use a mobile device's location to translate strokes on the screen into strokes in the real world for everyone to see. You hold your phone up and see pass through video from your camera. When you touch the screen, a combination of your location and orientation and your finger gestures will store strokes in a database to "draw". Someone who holds up the same app from a similar location will see the strokes that you input. Think of it as a simpler version of Tilt Brush, but in AR instead of VR. No computer vision would be required, just projection.
- Gunshot locator. Create devices that can precisely measure the time of arrival of a gunshot sound (maybe use a hammer or firecracker as a stand-in for testing). With 3+ such devices deployed in a

cluster at fixed known locations, it should be possible to triangulate to solve for the unknown origin of a loud noise.

Maybe less relevant or difficult to work with ideas

- Develop a few quantum algorithms that solve what would be intractable problems, but with a small N. Apply quantum computing to some new problem space.
- Real-time style transfer. Apply the style from a source image to the current frames of a cell phone camera.
- remote control for your remote control: create a miniaturized device that can be attached to your TV remote that will make a sound when you press a button from across the room. *(challenge: make it small. Probably not a good fit for CS, but Xin will pass to ECE and MAE.) Happy to pass this idea.*
- solar-powered, software-controlled wood burning kit. You would robotically move a magnifying glass along a path specified by a simple drawing program (which you would write). The wood would be positioned at the magnifying glass' focal length such that it would burn along the path. This would share much of the control concepts from DIY 3d printing rigs, with the added complexity that you'd need to find a way to keep it (or a mirror above it) pointed at the sun. (good embedded system project, perhaps difficult to pull off. MAE-CS joint)