Welcome!

https://cs342.stanford.edu

cardinalkit.slack.com

Don’t forget to record lecture
Dose of Digital Health

Kaiser signs 10yr cloud deal with Microsoft Azure
- Traditionally EHRs have been managed with on-prem servers
  - Lower latency
  - Thought to be more secure
  - Thought to be easier to maintain

Cloud is the future!

Don’t forget to record lecture
Welcome!

https://cs342.stanford.edu

cardinalkit.slack.com

Don’t forget to record lecture

Announcements

Santiago Gutierrez
Oliver Aalami
Mike Hittle
Varun Shenoy
Aish Venkatramani

Winter 2020
Assignment #3: App (alpha) for Midterm

Deliverables
For this assignment, each team will work together to create one app for their project. This is what you and your team will build upon for the remainder of the course. You will be graded for completion of the following:

App (alpha) for Midterm: Due February 11th

Grading Criteria: [link]

Visit cs342.stanford.edu for presentation rubric and template!
Overview for today

- Firebase Security
- Web Portals (ReactJS and others)
- Getting your apps ready for launch at Stanford
  - Guest lecture / Q&A with Garrick Olson and Lei Wang

We will be having OH today after class!

https://cs342.stanford.edu

cs342-aut1920.slack.com

Don’t forget to record lecture
Traditional Security

Server grants a token to the client. The client uses this token as an “ID” to verify identity and access. Need to write and maintain code (costly, time-consuming).
Firebase Security

Security made simple without the need to write or maintain code!
Firebase Security (Firestore)

Security made simple without the need to write or maintain code!
service cloud.firestore {
  match /databases/{database}/documents {

    match /users/jeffd23 {
      // Single document
    }

    match /users/{userId} {
      // Single Collection
    }

    match /posts/{postId=**} {
      // Recursive wildcard, includes all subcollections
    }
  }
}

MATCH
point to a
database location
service cloud.firestore {
  match /databases/{database}/documents {
    match /users/jeffd23 {
      allow read, write;
    }
  }
}
service cloud.firestore {
  match /databases/{database}/documents {
    match /users/jeffd23 /
    
    allow read: if request.auth !== null;
  }
}
service cloud.firestore {
  match /databases/{database}/documents {

    match /users/jeffd23 {

      allow read: if request.auth != null;

      allow write: if request.auth.uid == 'jeffd23';

    }
  }
}
service cloud.firestore {
  match /databases/{database}/documents {

    match /users/{userId} {

      allow read: if request.auth != null;

      allow write: if request.auth.uid == userId;

    }

  }

}
CS342 Security Rules for Firestore

The CardinalKit iOS app is pre-configured to work with these rules!

```python
rules_version = '2';

service cloud.firestore {
  match /databases/{database}/documents/studies/{studyId}/users {
    match /{userId}/document=** {
      allow read, write: if request.auth.uid == userId;
    }
  }
}
```
Are you interested on building a web-portal during CS342?

https://github.com/CardinalKit/CardinalKit-CS342-Web
“Backend” Architecture

Firebase iOS SDK

Cloud Identity Authentication

Google Cloud Healthcare API

Google BigQuery

Google Cloud Storage

Cloud Firestore

NoSQL Database

and more!
“Backend” Architecture

Firebase Web SDK

Cloud Identity Authentication

iOS SDK

Google Cloud Healthcare API

Google BigQuery

Google Cloud Storage

NoSQL Firestore

and more!
Possible Pathways for Building a Web App

*seem interesting? Take CS 147!
Possible Pathways for Building a Web App

*seem interesting? Take CS 147!
Tools

- **Goal**: build a simple, extensible architecture that can leverage CardinalKit collected data to do more *interesting* things
- **Client**: CardinalKit
- **Database/Security**: Firebase
- **Portal**: Vanilla JS, React, Django, etc.
- **Web Server**: Python, Go, Rust, C++, etc. (find a framework for HTTP calls) + AWS Lambda
  - Run analytics from web server (can query Firebase if needed!)
  - GCP also offers several tools for analytics and machine learning

*seem interesting? Take CS 147!"
Ideas

- Provide data visualization on the client side
  - Add a SwiftUI tab and query Firebase with Swift
- Interactive web portal visualizations
  - Use D3.js + vanilla HTML/JS to show data visualization with basic Firebase provider verification
- Use a Python stack to analyze collected data to compile a .pdf report
  - HTTP requests!
- etc.

*seem interesting? Take CS 147!*
Set a document

To create or overwrite a single document, use the `set()` method:

```javascript
// Add a new document in collection 'cities'
db.collection('cities').document('LA').set({
  "name": "Los Angeles",
  "state": "CA",
  "country": "USA"
})
```

---

Set a document

To create or overwrite a single document, use the `set()` method:

```javascript
// Add a new document in collection 'cities'
db.collection('cities').doc('LA').set({
  name: "Los Angeles",
  state: "CA",
  country: "USA"
})
.then(() => {
  console.log('Document successfully written!');
})
.catch((error) => {
  console.error('Error writing document: ', error);
});
```

---

CONNECT TO DATABASE

```javascript
// Connect to the database
db = firebase.database()
```

---

WEB APP

```javascript
// Web app
```

---

SWIFT

```swift
// Swift
```

---

OBJECTIVE-C

```objective-c
// Objective-C
```

---

JAVA

```java
// Java
```

---

KOTLIN

```kotlin
// Kotlin
```

---

JAVA

```java
// Java
```

---

PYTHON

```python
// Python
```

---

C++

```cpp
// C++
```

---

NODE.JS

```javascript
// Node.js
```

---

MORE

```
```

---

© 2016 Stanford Byers Center for Biodesign
Firebase Security

Security made simple without the need to write or maintain code!
Firebase Security for web (Firestore)

Your rules for the web might look a little bit different!

Here are some that CardinalKit recommends as a starting point. First, create an admin collection with each authorized administrator.

```
/studies/edu.stanford.cardinalkit/admin/santig@stanford.edu

<table>
<thead>
<tr>
<th>edu.stanford.cardinalkit</th>
<th>admin</th>
<th><a href="mailto:santig@stanford.edu">santig@stanford.edu</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Start collection</td>
<td>+ Add document</td>
<td>+ Start collection</td>
</tr>
<tr>
<td>admin</td>
<td><a href="mailto:santig@stanford.edu">santig@stanford.edu</a></td>
<td>Add field</td>
</tr>
<tr>
<td>users</td>
<td></td>
<td>enabled: true</td>
</tr>
</tbody>
</table>
```
Firebase Security for web (Firestore)

Then, update your rules to grant access to admins

```
rules_version = '2';

service cloud.firestore {
    match /databases/{database}/documents/studies/{studyId}/users {
        match /{userId}/{documentId} {
            allow read, write: if isUserLoggedIn(userId) || isAdmin(database, studyId);
        }
    }

    function isUserLoggedIn(userId) {
        return request.auth.uid == userId;
    }

    function isAdmin(database, studyId) {
        return ("email" in request.auth.token)
        && isAdmin(database, studyId) != null
        && isAdmin(database, studyId).data.enabled == true;
    }

    function getAdmin(database, studyId) {
        return get(/databases/${database}/documents/studies/${studyId}/admin/${request.auth.token.email})
    }
```
IMPORTANT

Be prepared to update your rules to work with (1) Cloud Storage and/or (2) Stanford University

Rules are very similar to Firestore!
[documentation]
University Managed Model:

- Defines how services can communicate with each other, securely, so you don’t have to.
- Already understands your system, which means your studies get expedited DRA review.
- Helps you get started with premium GCP services at a low-cost — and helps you scale up when needed.
CS342 final presentation:

- your apps should work under a Stanford-managed cloud!
- rules may be different

GoogleService-Info.plist
Overview

- **Getting your apps ready for launch at Stanford**
  - Guest lecture / Q&A with Garrick Olson and Lei Wang

We will be having OH today after class!

https://cs342.stanford.edu

cs342-aut1920.slack.com

*Don’t forget to record lecture*
## Midterm Rubric

<table>
<thead>
<tr>
<th>Project Background</th>
<th>Need Statement</th>
<th>Informed Consent</th>
<th>Surveys</th>
<th>Tester Feedback</th>
<th>Next Steps</th>
<th>Presentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding of research/background and chosen path. Clear reference to a proven need.</td>
<td>Clarity in chosen need area and population and outcome.</td>
<td>Consent from users and patients, for surveys and testing.</td>
<td>Reporting of patient outcomes and ResearchKit use.</td>
<td>How was the research done to receive feedback? What are key understandings from the feedback?</td>
<td>Are the next steps feasible? Name all that apply: Health Records, HealthKit (fitness data), CareKit (medication adherence), Web Portals, Visualization, etc.</td>
<td>Every student participated, clear communication of ideas and concepts.</td>
</tr>
</tbody>
</table>

(15 points) (10 points) (15 points) (15 points) (15 points) (20 points) (10 points)
Attendance Check 👍🏼

- Use the following link to mark your attendance for today:

[https://cs342.stanford.edu](https://cs342.stanford.edu)
Thank you!
STANFORD BYERS CENTER FOR
BIODESIGN