**A BOOK DATABASE**

Enter the books you own

Enter the books you’d like to buy

Store the book info, images

**A deal finder** (suggested implementation: **web app with mobile notifications**).

Build a simple web app to notify you when an item you covet goes on sale for a good price.

You could use a web scraper to pull the item’s product page and notify you of any price changes.

**An expense tracker** (suggested implementation: web or mobile app).

Create a simple interface you can use to add and categorize your expenses.

Generate monthly reports based on the inputs and write custom alerts for things, like, “spending too much money on coffee… as always.”

**A financial independence calculator** (suggested implementation: web app).

Financial independence is, essentially, saving and investing as much of your income as possible

so that you don’t need to work for money.

While many financially independent people continue to work,

they can now focus on doing work that they love, rather than work that pays the most.

Build a tool to calculate, based on your: current savings, investments, income, retirement accounts, and expenses,

how far away you are from financial independence.

Some examples: FIREcalc, cFIREsim.

https://www.firecalc.com/

http://www.cfiresim.com/

**A movie showtime finder** (suggested implementation: **web or mobile app with email or text message notifications**).

Build a program that notifies you, by text or email,

about showtimes for potentially interesting movies playing at your favorite cinema.

The concept of an 'interesting movie' can be derived using machine learning (if you watch enough movies to have good training data),

or a handcrafted algorithm. For example, you might use the Open Movie Database API,

paired with an HTML parser like BeautifulSoup,

to build a program that alerts you to sci-fi movies rated 7.0 or above on IMDB,

movies starring Amy Adams, and/or any movie with an average rating of 8.0 or higher.

**Pixel art generator** (suggested implementation: any programming language you want to master).

Build a tool that takes an image as input and samples the image to produce pixel art as output.

If you want to improve your front-end skills, generate the resulting pixel art using CSS.

**Microlearning app** (suggested implementation: **web or mobile app**).

Build an application that sends you one page per day about something you want to learn.

This could be a random page from Wikipedia, a page of React documentation, a Kanji character, or a page from the CIA World Factbook

**Daily desktop background** (suggested implementation: **Unsplash API, scripting language for your OS**).

Build an app that refreshes your desktop background with a new image every day

**Lunch picker** (suggested implementation: **CLI, web or mobile app**).

If you’re a working software engineer, you’ve probably wrestled with one of the toughest questions in software development…

where should I have lunch? Your lunch picker is the tool you’ll turn to to answer this question.

It can be super simple and pick from a range of options you know you like at random,

or more complex — pulling in data from Google Reviews and taking into account: location, price, and type of cuisine.

**Date planner** (suggested implementation: **mobile app**).

One of the hardest things about dating,

or being in a long-term relationship,

is deciding where to go on dates.

Build a tool that scours restaurant reviews, event calendars, and other data for date idea suggestions

**‘Bring your umbrella’ notifier** (suggested implementation: **mobile app**).

Build a simple mobile app that sends you a phone notification in the morning

telling you if you should bring your umbrella to work

(e.g., if it is going to rain in your area that day).

**Create and automatically update a data set** (suggested implementation: **web scraper input to CSV output**).

Build a tool to automatically build and update a data set about something you’re interested in.

Suggested data sets: stats about your favorite sports team,

flights to destinations you want to visit,

meteorological data from where you live,or anything else that interests you.

The most important thing is that the data set should require constant updates as new data is generated — and should occur automatically.

For example, as soon as your sport’s team’s results are posted on a website,

the data should be automatically scraped and added to your data set.

**An Apartment Manager Application (web and mobile application)**

An application that manages and tracks properties for landlords

Manages posts to social media to aid in the marketing of rental properties

Track property issues

Help tenants find rental properties

**Platforms:**

Java

Android

HTML

CSS

Javascript

**Database:**

MySQL

Oracle

**Web Server:**

Apache Tomcat

Websphere

**Message Queue:**

Sender and receiver of the message do not need to interact with the message queue at the same time. Messages placed onto the queue are stored until the recipient retrieves them.

IBM MQ

Oracle Advanced Queuing

Java Message Service

Amazon Simple Queue Service (cloud)

Strom MQ (cloud)

Iron MQ (cloud)

Apache ActiveMQ (opensource)

Apache Kafka (open source)

Apache Qpid (open source)

**Application Features:**

1. **Landlord**
   1. Landlord Profile
      1. Name
      2. Email
      3. Contact Number
   2. Post Property
      1. Pictures
      2. Address
         1. Street number
         2. City
         3. Region
         4. ZIP Code
      3. Map Location
         1. Maps integration
         2. GPS
         3. Latitude Longitude
      4. Cost
      5. Terms
      6. Occupied or Unoccupied status
      7. Property type
         1. Retail
         2. Commercial
         3. Residential
      8. Other Details
      9. Post to social media or wherever
      10. Landlord Contact Details
          1. Email
          2. Phone
   3. List Apartments (Add filters)
      1. Occupied
      2. Unoccupied
      3. Needs Maintenance
      4. Past due payments
   4. Manage Apartments
      1. Assign Tenants
      2. Send notice to tenants
      3. Receive messages from feelers
      4. Apartment Maintenance
         1. Details on repairs and maintenance
         2. Cost
         3. Track security deposit based on maintenance and damages
      5. Collect Payment
         1. Payment gateway?
         2. Track payments
         3. Send payment reminders
   5. Receive Alerts
      1. Managed Apartment
         1. Need maintenance
         2. Collect Payment
         3. Find tenant
         4. Tenant contract is up
         5. Remind other activities to be performed
      2. Posted Apartment
         1. Remind that the apartment has no tenants and suggest actions
         2. Post to social media every 2-3 days
         3. Look for tenants
         4. Update apartment details
   6. Property report
      1. Portfolio
      2. Apartment
2. **Tenant**
   1. Tenant Profile
      1. Name
      2. Email
      3. Contact Number
   2. Find Apartments
      1. Can be anonymous user
      2. Filters
         1. Car park
         2. Bedrooms
         3. Size
         4. Price range
         5. Within Radius of marker
         6. Within City
      3. List Apartments
         1. List
         2. Marker
      4. Send contact request to landlord
   3. Rent Apartment
   4. Pay Apartment
      1. Payment gateway?
      2. Track payments
   5. Request maintenance
      1. Details on repairs and maintenance
      2. Cost
      3. Track security deposit based on maintenance and damages
   6. Receive Alerts
      1. Payments
      2. Landlord notices
3. **Admin**
   1. BAN Tenant
   2. BAN Landlord