Web Catechism

Web Catechism is going to be a responsive, performant Web App that displays the content of the New City Catechism (NCC). The NCC's copyright encourages distribution with certain stipulations by which I will abide.

The current Web App is written is not responsive to different viewport sizes, does not make use of any localStorage or cookies to store user preferences, and (in my opinion) has an awkward navigation scheme. I'd like to improve on those aspects by creating an intuitive navigation system and using localStorage to cache preferences. I'd also like to create a Login system that allows users to sync preferences and progress across devices.

Web Framework

Web Catechism will be creating using HTML, CSS (SCSS), and JavaScript (both on the client and server side). I'll use Node.js as the Web Framework, with various libraries to extend its capabilities and features. Node.js is an event-driven, asynchronous JavaScript platform that uses Chrome's V8 engine to run on server-side hardware. It's been used with great success at major corporations and small businesses alike.

Features & Schedule

This application will load and display a single section of the catechism (one question, answer, scripture, and video) at a time. It will have intuitive navigation to reach the previous and next questions, and a way to navigate to a specific question without just using the "previous" and "next" options.

Version 0 (Week 1):
- research and analysis on underlying technology stack (including Node.js and MongoDB)
- design UI (moodboards, wireframes)
- publish PRD (this document)

Version 1 (Week 2):
- create basic App structure through MVC framework

- create functioning navigation and UI
- write content (52 questions, answers, scripture passages, and videos)

Version 2 (Week 3):
- continue UI refinements
- implement localStorage progress saving for all users
- implement deferred loading for video

Version 3 (Week 4):
- continue UI refinements
- create login functionality
- attach cross-device progress saving / restoration to login functionality

Database UML

Since I'm building this application with Node.js, I'll be using a non-relational document-oriented database called MongoDB. I'll have one collection (table) with as many documents (rows) as I have users. The minimum viable fields (columns) are detailed below:

Users Collection

```
{
    "_id": ObjectId("12 byte uuid"),
    "user_name": "email@example.com",
    "password": "bcrypt",
    "progress": 16
}
{
    "_id": ObjectId("12 byte uuid"),
    "user_name": "different@example.com",
    "password": "bcrypt",
    "progress": 10
}
```

Moodboard

Colors

Typography (League Gothic and Raleway)

Heading Level 1 (3.998em)

Heading Level 2 (2.872em)

Heading Level 3 (1.999em)

Body copy is set in Raleway @ 1.5em. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

User Interface

Clickable Header

Prev   Next

Link Text

Wireframes

See All Questions   login or signup

New City Catechism

Part 2

Christ, redemption, grace

Prev   Next

Question 21

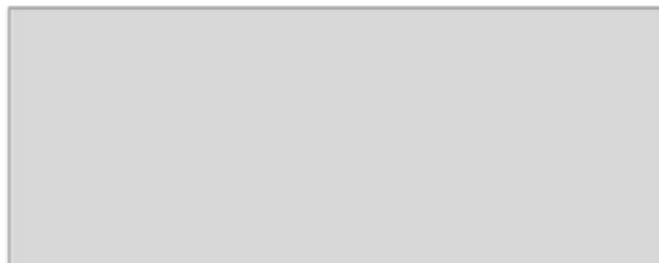What sort of Redeemer is needed to bring us back to God?

Answer

One who is truly human and also truly God.

Scripture

> For to us a child is born, to us a son is given; and the government shall be upon his shoulder, and his name shall be called Wonderful Counselor, Mighty God, Everlasting Father, Prince of Peace.
> —Isaiah 9:6 (ESV)

Video

New City Catechism

Part 3

Spirit, restoration, growing in grace

Question 36

What do we believe about the Holy Spirit?

Answer

That he is God, coeternal with the Father and the Son, and that God grants him irrevocably to all who believe.

Scripture

For none of us lives to himself, and none of us dies to himself. For if we live, we live to the Lord, and if we die, we die to the Lord. So then, whether we live or whether we die, we are the Lord's.
—Romans 14:7–8 (ESV)

Video