Project Plan
Dark Theme Darkening

The Capstone Experience

Team Mozilla
Vivek Dhingra
Zhengyi Lian
Connor Masini
Bogdan Pozderca
Dylan Stokes

Department of Computer Science and Engineering
Michigan State University
Spring 2018
Functional Specifications

• Expand the Firefox theming API to support theming of additional parts of the browser
  ▪ Bookmarks bar, arrow panel, page-loading indicator

• Increase Google Chrome compatibility
  ▪ Add compatibility for Chrome only theming features to allow for the use of Chrome Themes in Firefox

• Update the Firefox Dark Theme to use the expanded API.
  ▪ Add dark coloration for bookmarks, menus, etc.
Design Specifications

• Users are less likely to switch browsers once personalized
• New theming API will allow developers to create more immersive themes
• In-Product pages
  ▪ Browser settings pages need theming for icons, background, etc.
• Menus
  ▪ Toolbars and select dropdowns are able to be themed
• Autocomplete history pop-up
  ▪ Customizable text and background colors
• Theme Transitions
  ▪ Current state is jumpy and disrupting
  ▪ Users should see a smooth transition between themes
Screen Mockup: In-Product Page
Screen Mockup: Bookmarks Bar
Screen Mockup: Menus

Is "too much internet" a real thing? We explore this idea by talking about what technology is doing to our bodies and brains in Mozilla’s latest IRL podcast. Tune in to learn more.
Technical Specifications

• Use JavaScript and CSS to expand theming API
  ▪ Compatible with Chrome and Firefox theme formats
  ▪ Themes are comprised of a JavaScript file and a JSON file

• Use Mercurial to upload files to Bugzilla review board
  ▪ Changes are reviewed by Firefox contributors
  ▪ Once review is passed, changes are pushed to the Nightly version of Firefox

• Use artifact builds to reduce compilation time
  ▪ Download pre-built C++ components rather than building them locally
  ▪ Trade bandwidth for time
System Architecture
System Components

• Hardware Platforms
  ▪ Ubuntu Linux Test System
  ▪ Microsoft Windows Test System
  ▪ Apple MacOS Test System

• Software Platforms / Technologies
  ▪ Mercurial for version control
  ▪ JavaScript and CSS to implement API
  ▪ Bugzilla and MozReview for code review
  ▪ Atom text editor
Risks

• **Large Repository**
  - **Description:** Firefox is over 35 million lines code, finding a place to start is challenging.
  - **Mitigation:** Using the Searchfox web tool to locate files of interest, rather than grep.

• **Platform Testing**
  - **Description:** Need to efficiently code for all platforms without breaking compatibility.
  - **Mitigation:** Write unit tests using Mozilla’s testing suite / physically test on all platforms.

• **Cross-Browser Themes**
  - **Description:** Compatibility when transitioning themes from Google Chrome to Firefox
  - **Mitigation:** Review resources to get a comprehensive understanding of Google Chrome themes.

• **Code Optimization**
  - **Description:** Need to ensure that any additions that are made do not slow down the browser and meet Mozilla’s coding standards.
  - **Mitigation:** Run unit testing on any added code that tests running times against thresholds and submit code for review.
Questions?