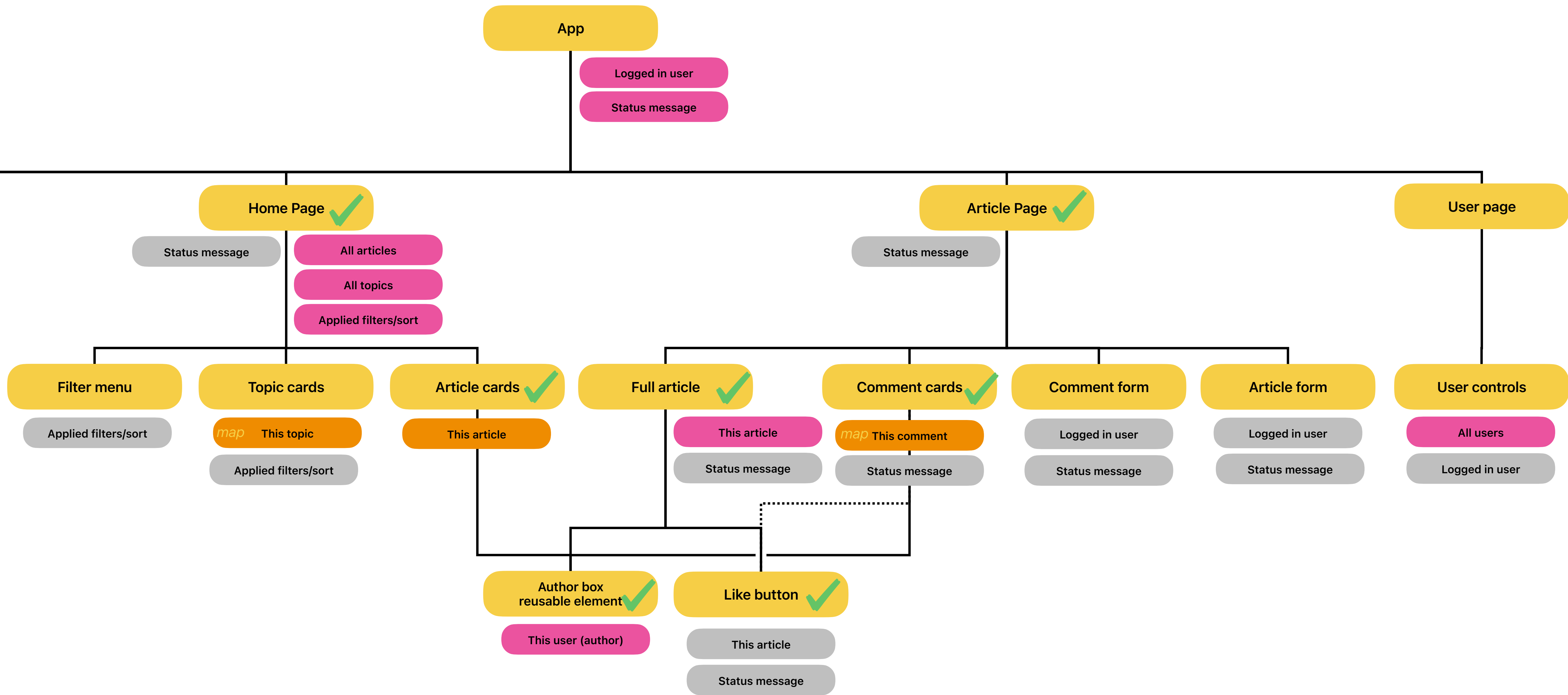


# COMPONENT TREE WITH STATE



| Branch Name              | Description  |
|--------------------------|--|
| 01-project-setup         | Project Initialised  |
| 02-cors                  | Cross Origin Resource Sharing (CORS) enabled   |
| 03-planning              | Project plan including wireframes for each route and component tree with associated state  |
| 04-all-articles          | Main page to display all articles  |
| 05-individual-article    | Article page to display individual article   |
| 06-view-comments         | Articles display associated comments   |
| 07-article-votes         | Enables voting for articles  |
| 08-post-comments         | Enables commenting on articles   |
| 09-delete-comments       | Enables users to delete their own comments   |
| 10-articles-by-topic     | Topics page to display articles filtered by topic  |
| 11-articles-sorting      | Enables sorting of articles either ascending or descending by date, comment count or votes |
| 12-error-handling        | User friendly error messages are shown for 404s and submitting incomplete forms            |
| 13-project-deploy        | Project published  |
| 14-read-me               | Read me added  |
| 15-styling-update        | Styling checks   |
| 16-responsiveness-update | Mobile & desktop display checks  |
| 17-accessibility-update  | Accessibility checks   |
| 18-ux-update             | UX checks  |

| Endpoints                                 |
|---|
| "GET /api/topics"                         |
| "GET /api/articles" with overrides        |
| "GET /api/articles/:article_id"           |
| "POST /api/articles"                      |
| "PATCH /api/articles/:article_id"         |
| "GET /api/articles/:article_id/comments"  |
| "POST /api/articles/:article_id/comments" |
| "PATCH /api/comments/:comment_id"         |
| "DELETE /api/comments/:comment_id"        |
| "GET /api/users"                          |
| "GET /api/users/:username"                |

**Backend Adjustment Required**

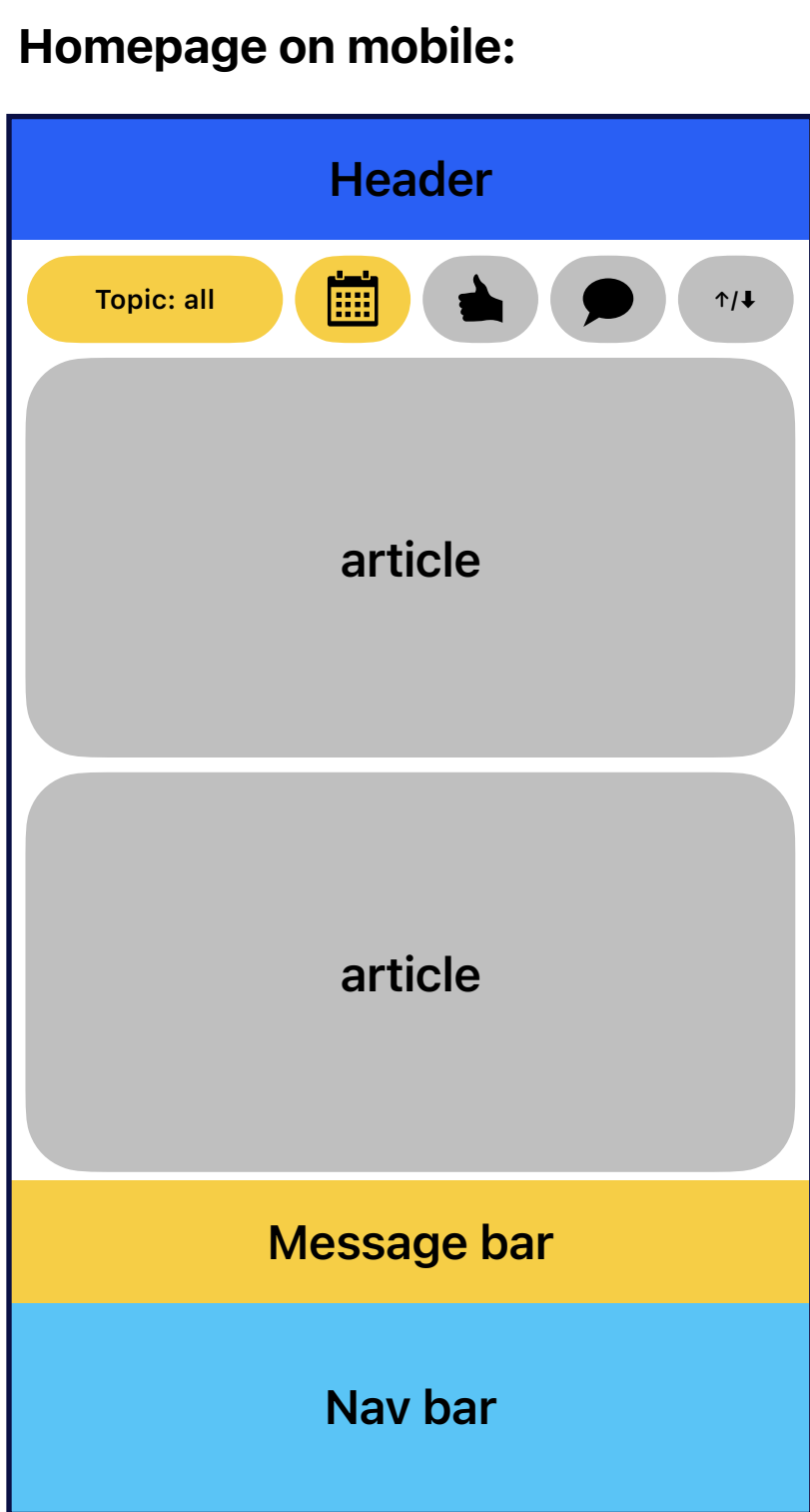
Planned feature: Some way of recording what users have liked which articles and comments so they can't add multiple likes, and can revoke their like on a future visit.

Probably achieved by recording the usernames of those who've liked a specific comment or article in a many-to-many joint table.

## PAGES



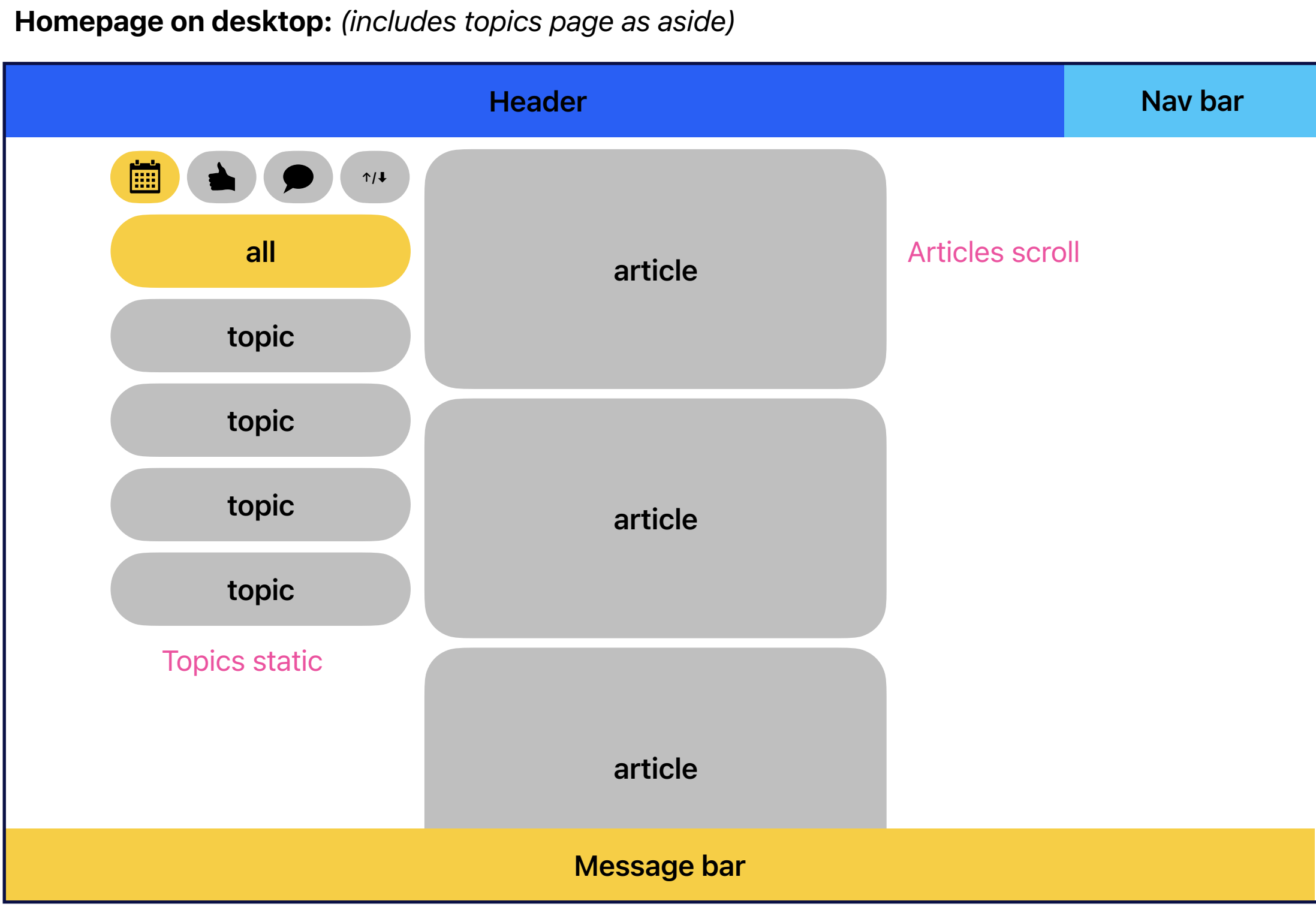
/ or /topic/:topic



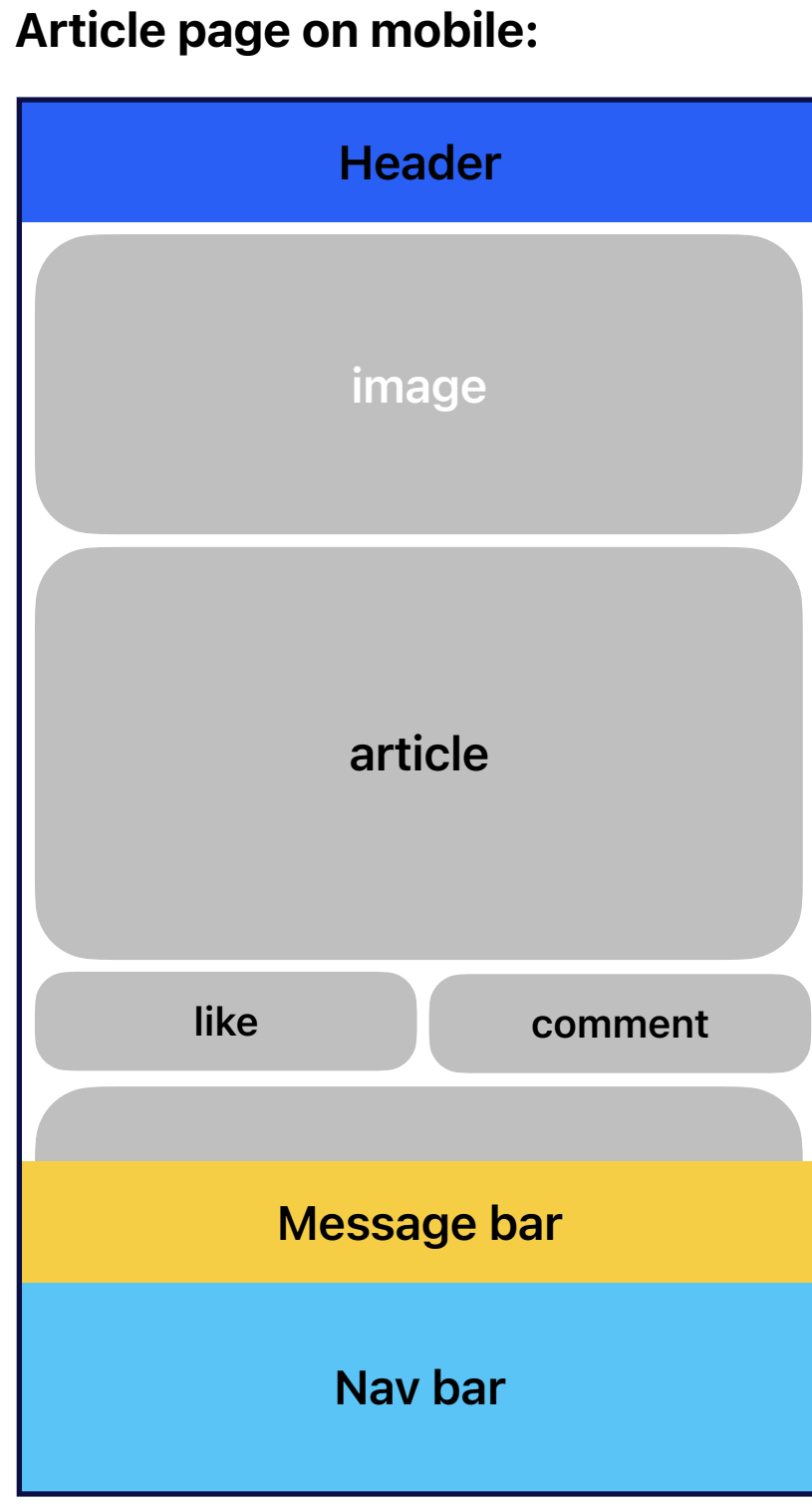
/topic/



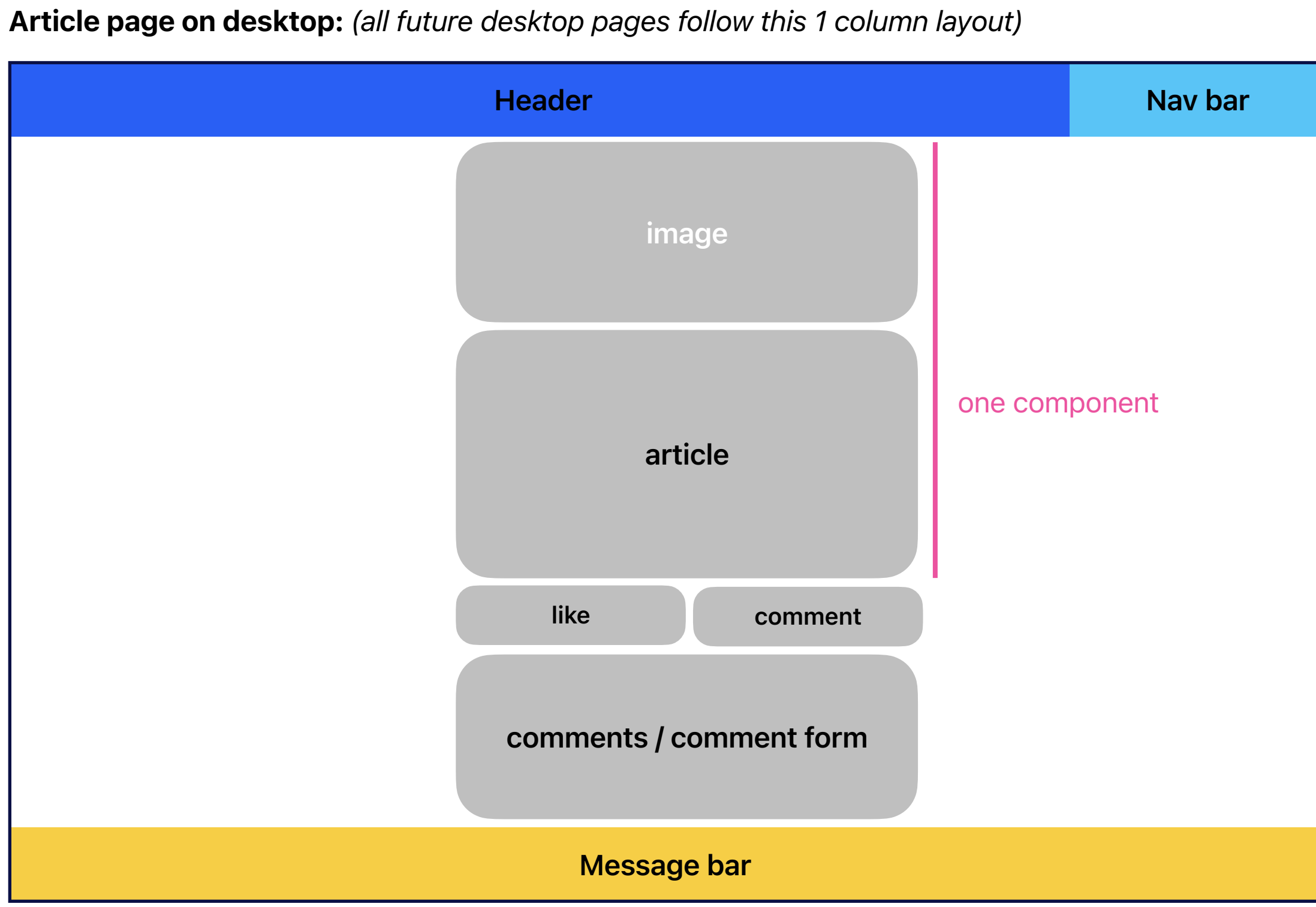
/ or /topic/ or /topic/:topic



/article/:article\_id



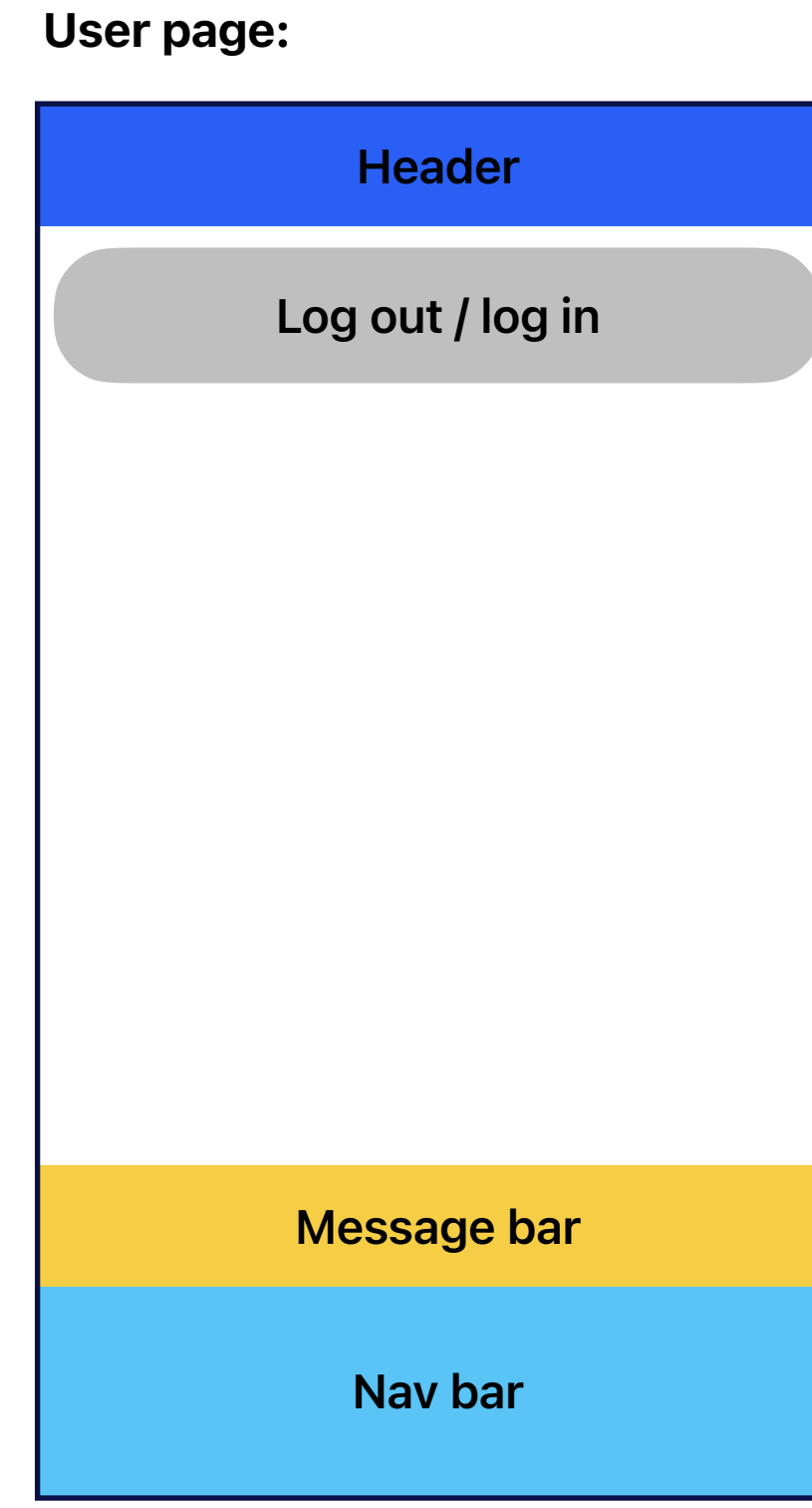
/article/:article\_id



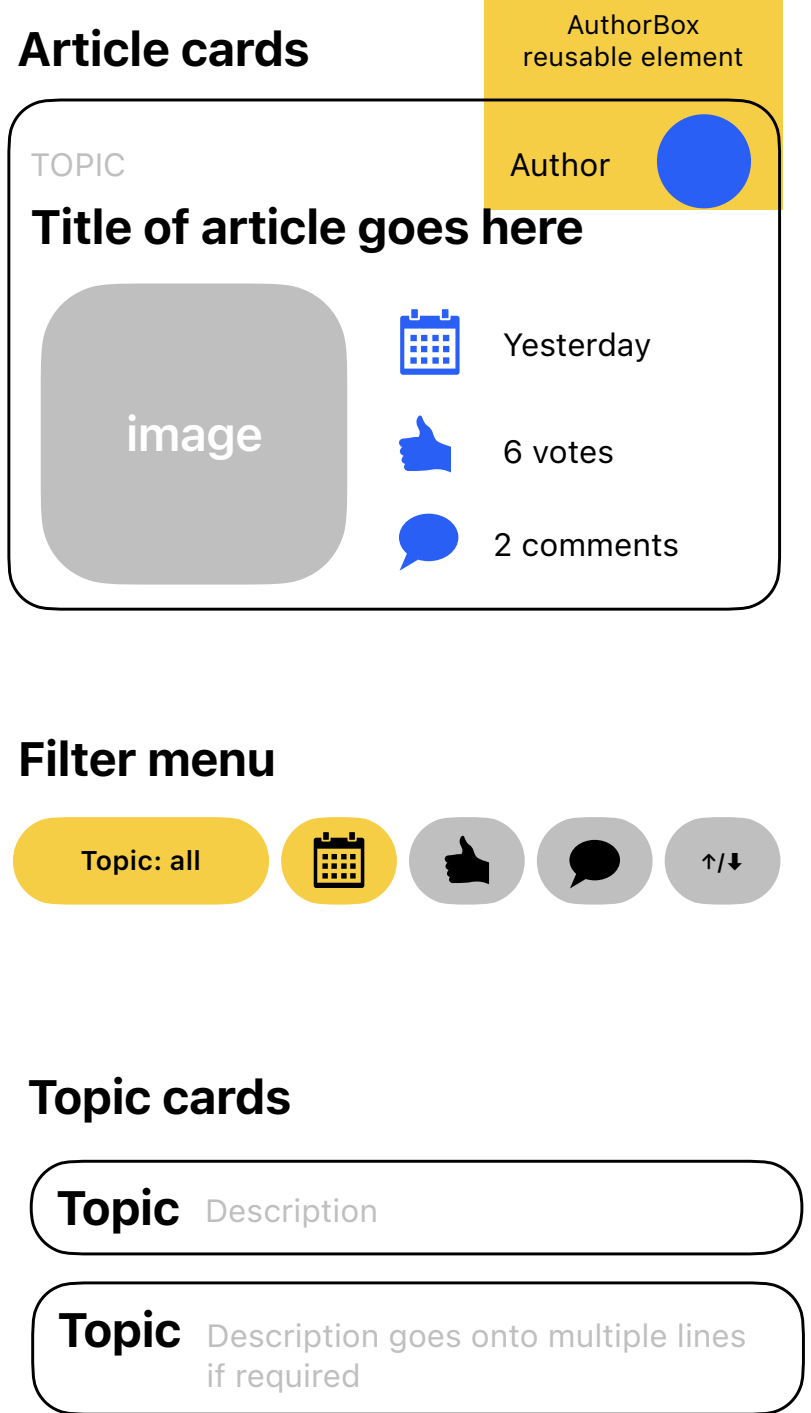
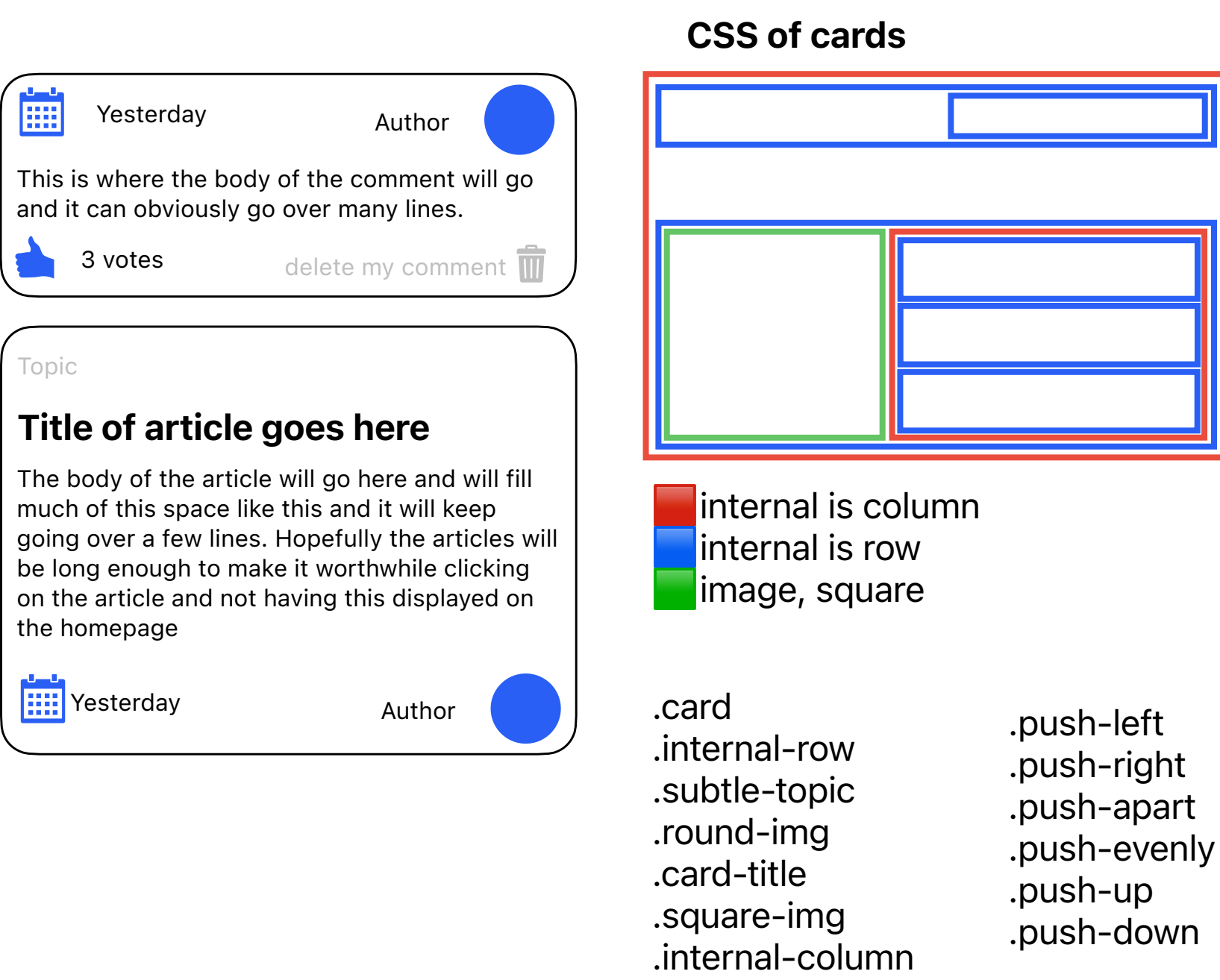
/article/



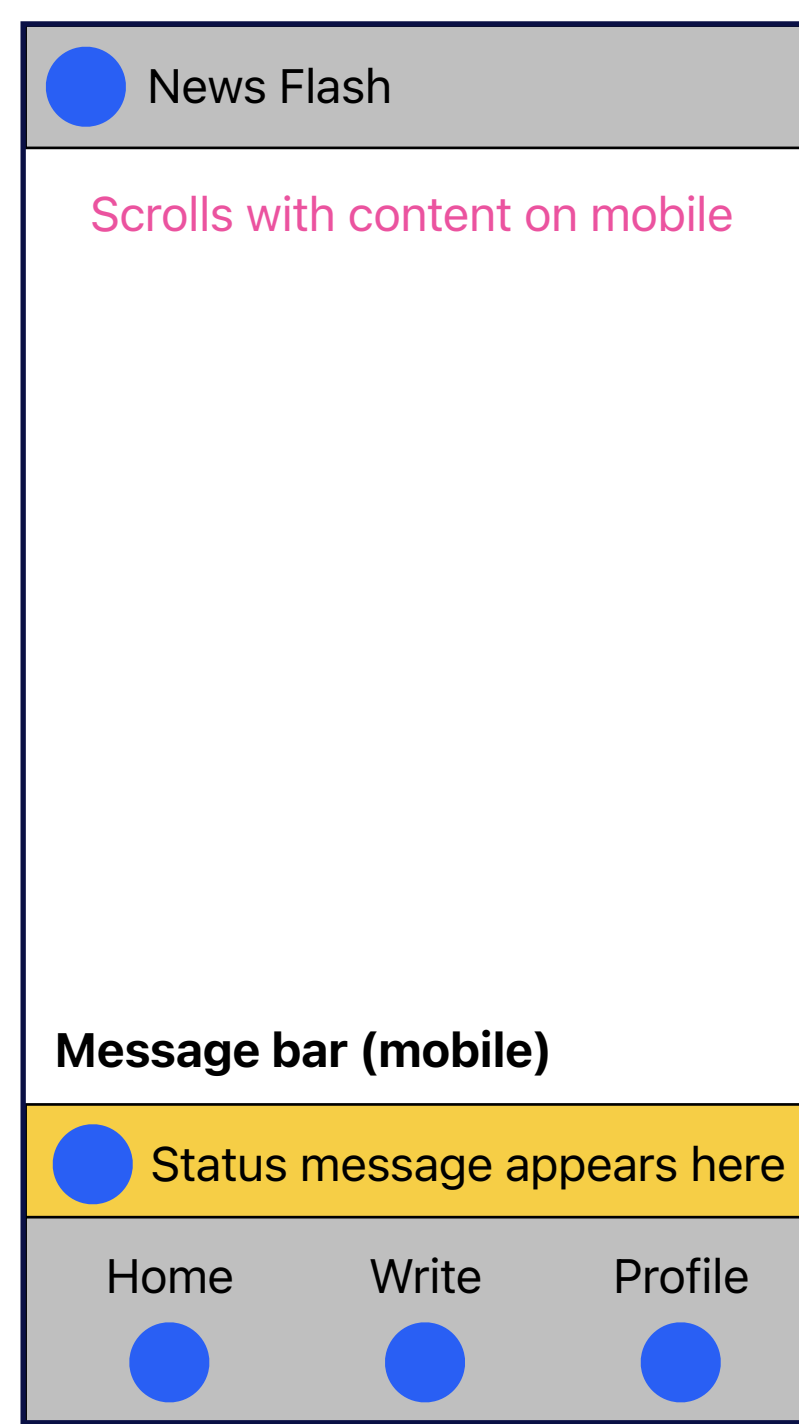
/user/



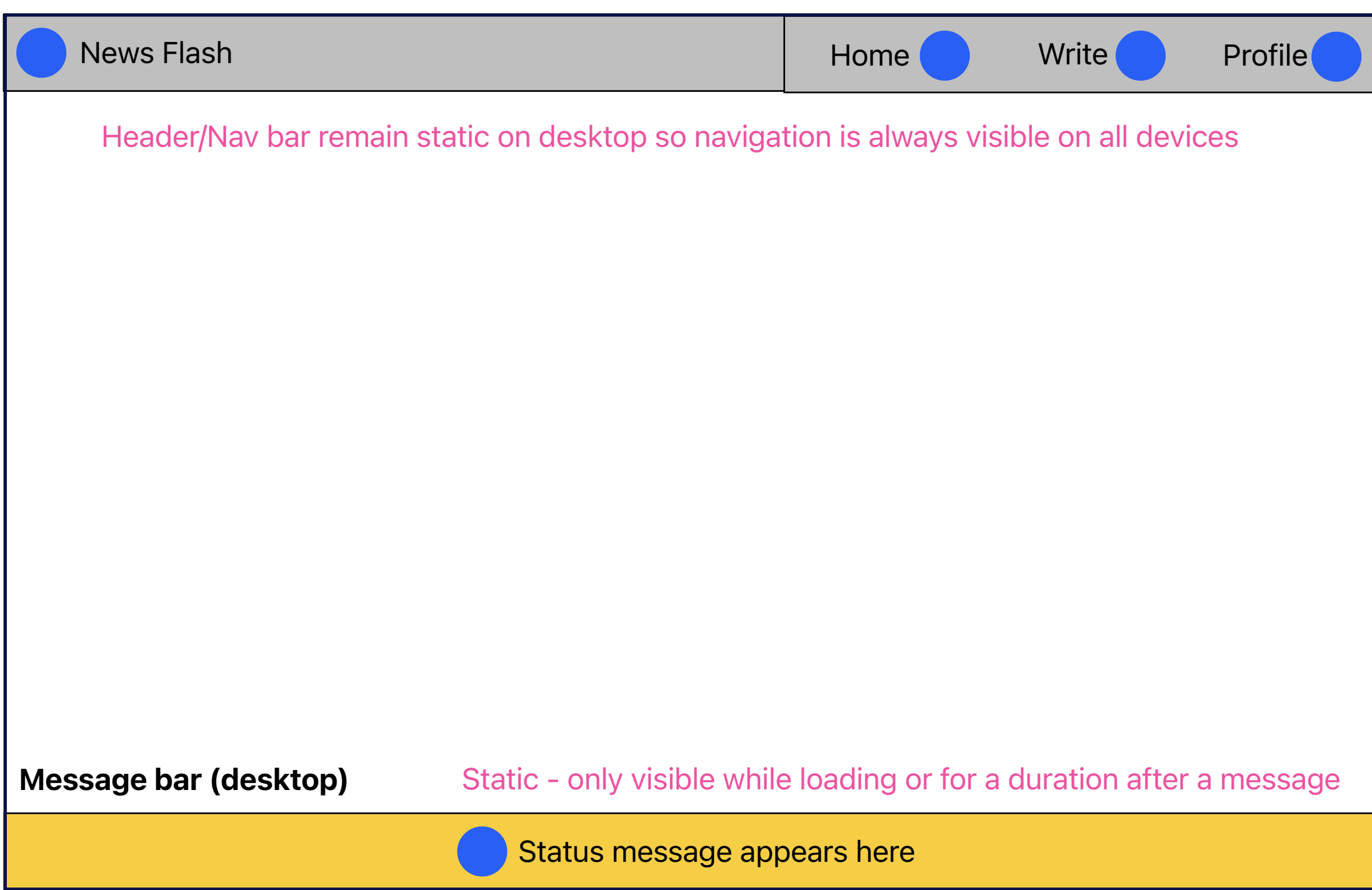
## COMPONENTS



Header (mobile)

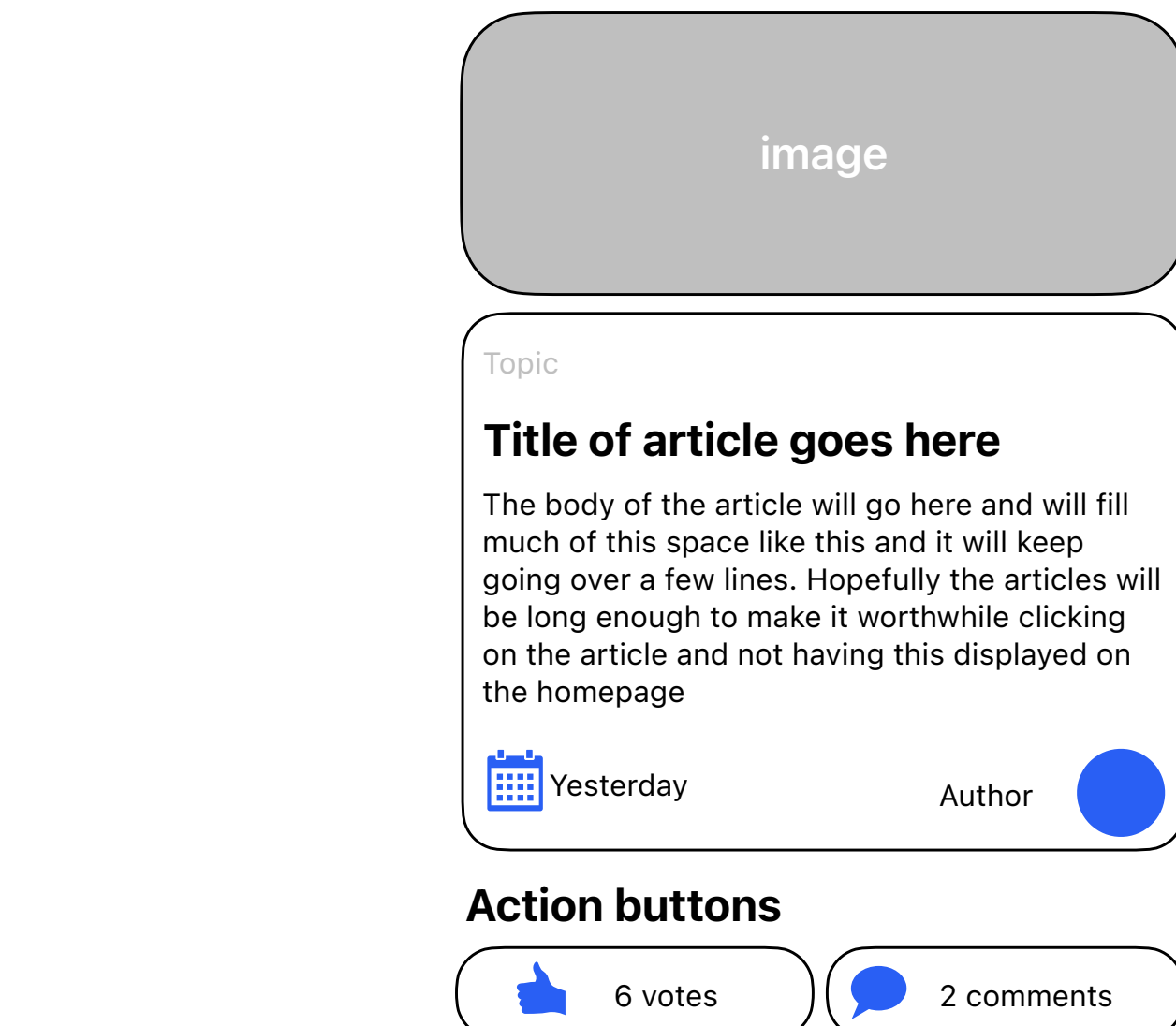


Header (desktop)

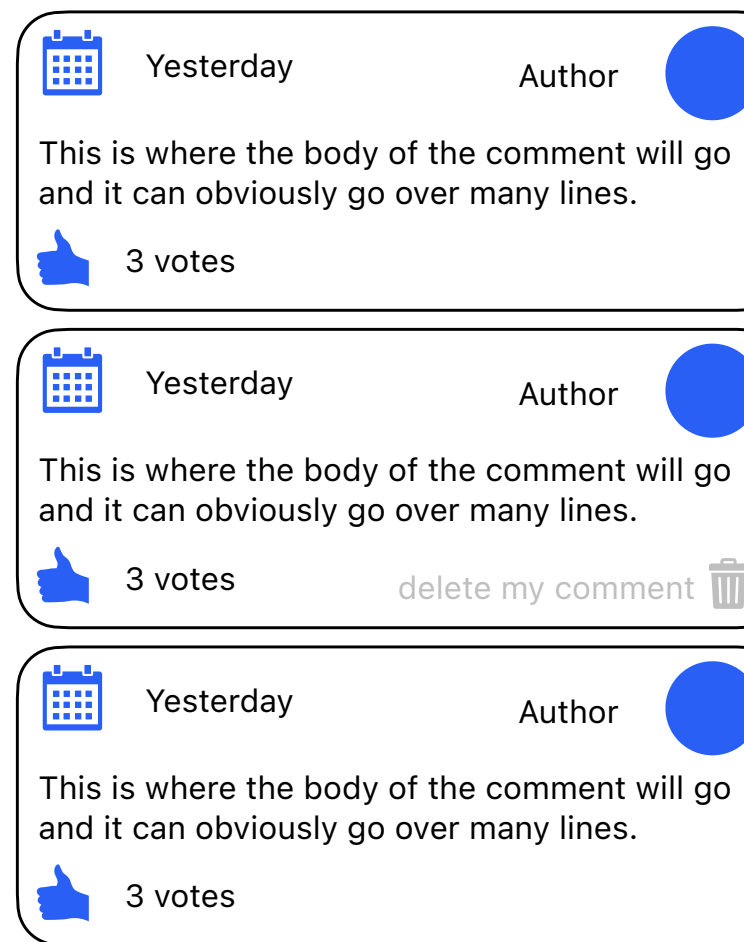


Navigation bar (desktop)

Full article



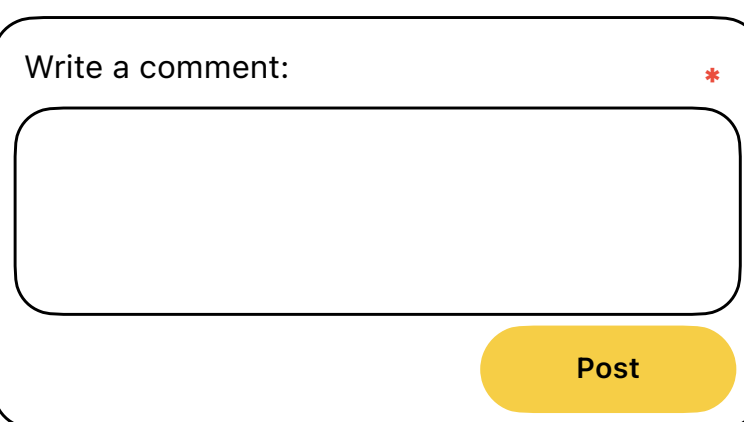
Comment cards



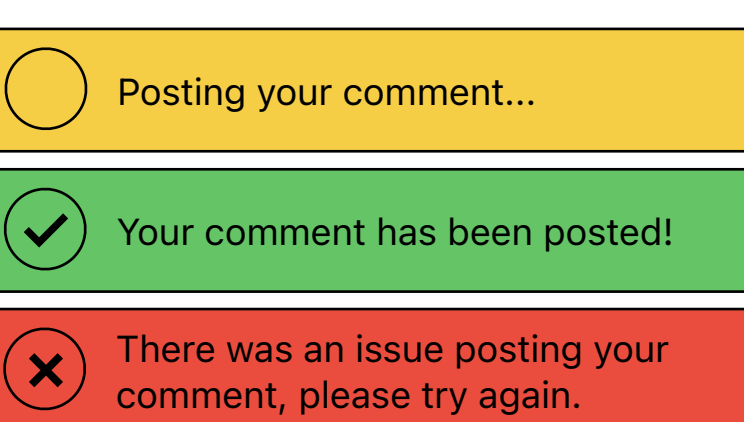
Relevant status messages



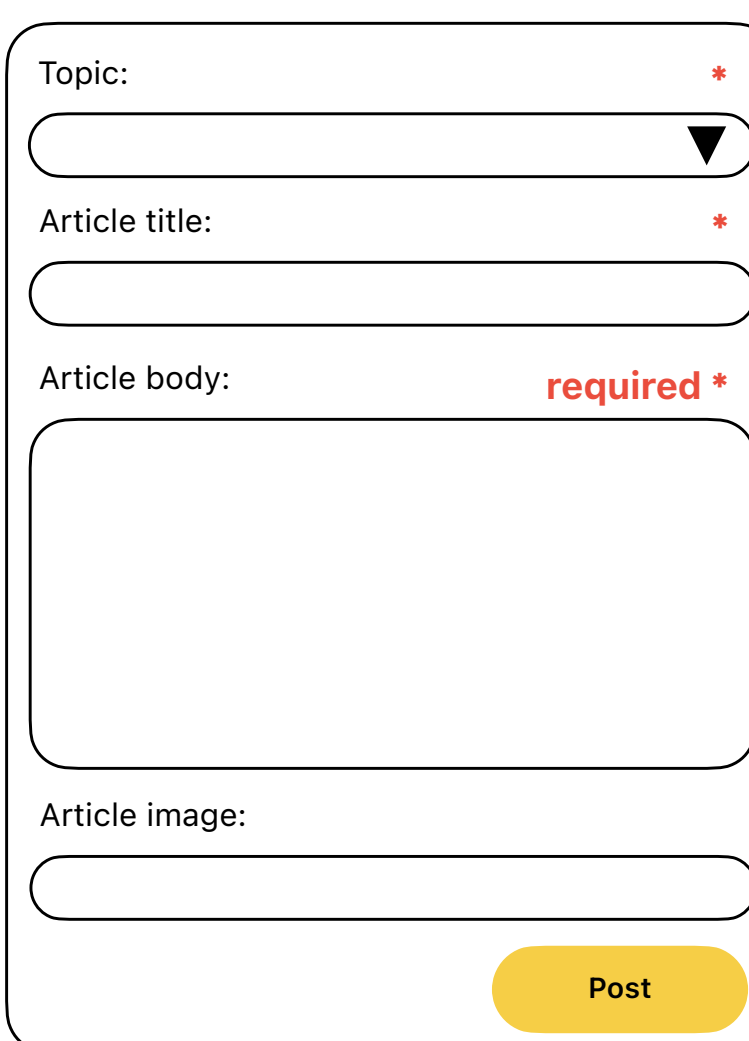
Comment form



Relevant status messages



Article form



Relevant status messages



red asterisk changes to grey when valid

"required" word appears if blur occurs and value is invalid