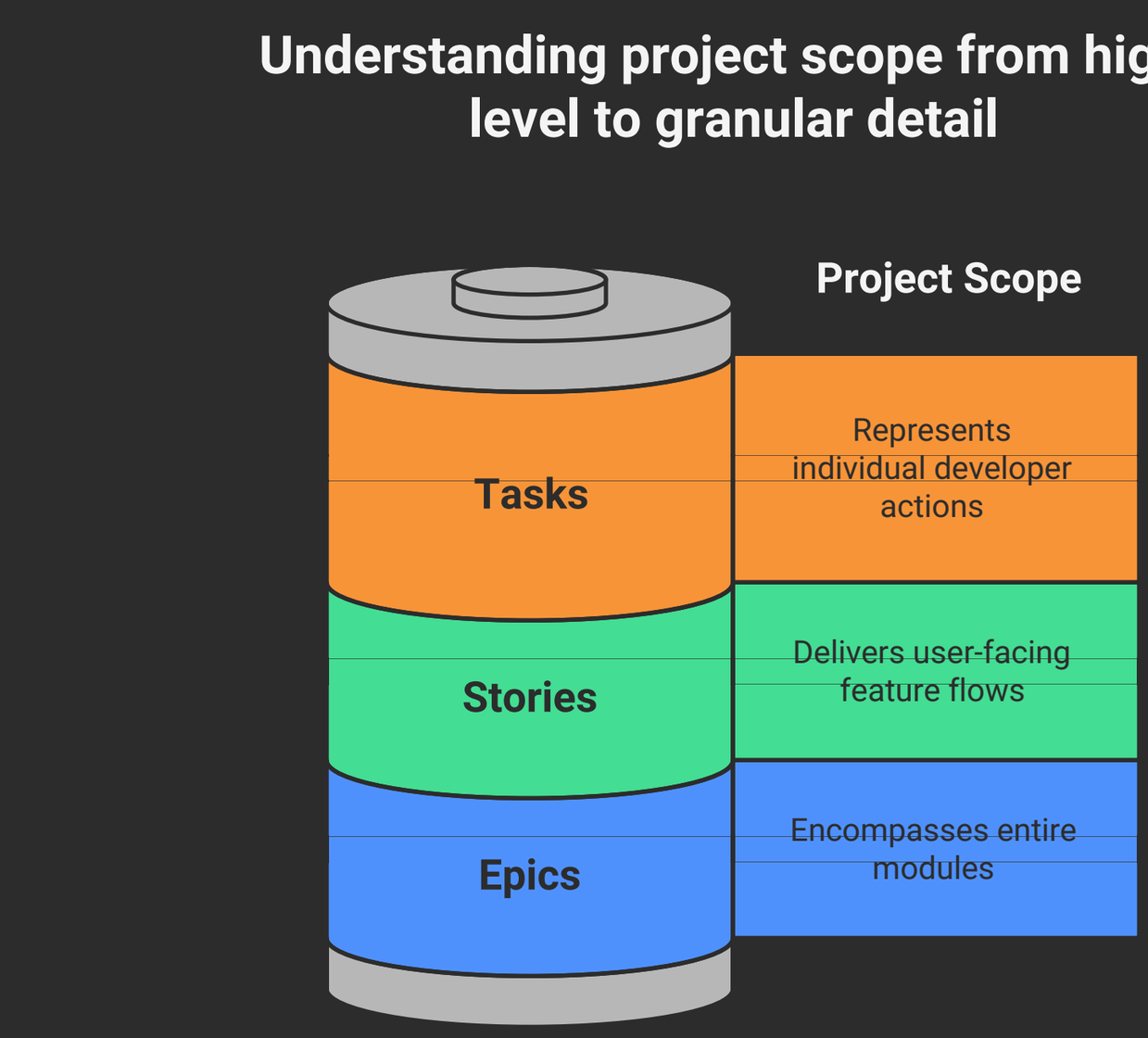
Expanded Agile Breakdown for StackIt – Feature Stories, Tasks & Epics

Agile Planning Structure

We'll organize this into:

- **Epics** = Large features/modules
- **Stories** = User-facing flows
- **Tasks** = Developer actions
- = includes ML/DL
- = critical MVP

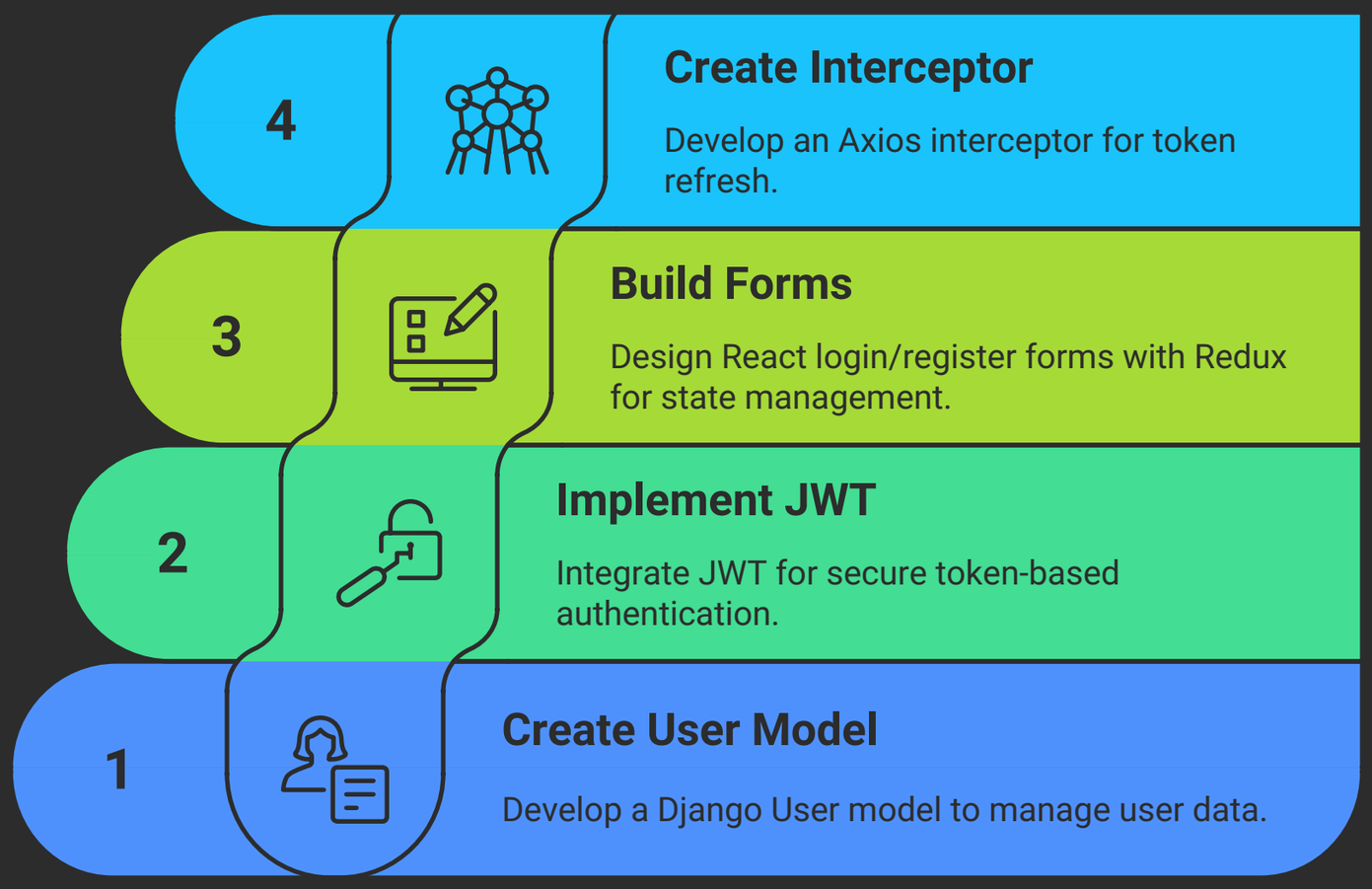


Epic 1: Authentication & User Management

StoryTasks

As a user, I can register and log in securely | - Create Django User model (extend if needed) - Implement JWT via DRF SimpleJWT - Build React login/register forms + Redux auth slice - Create token refresh interceptor in Axios

Secure User Authentication



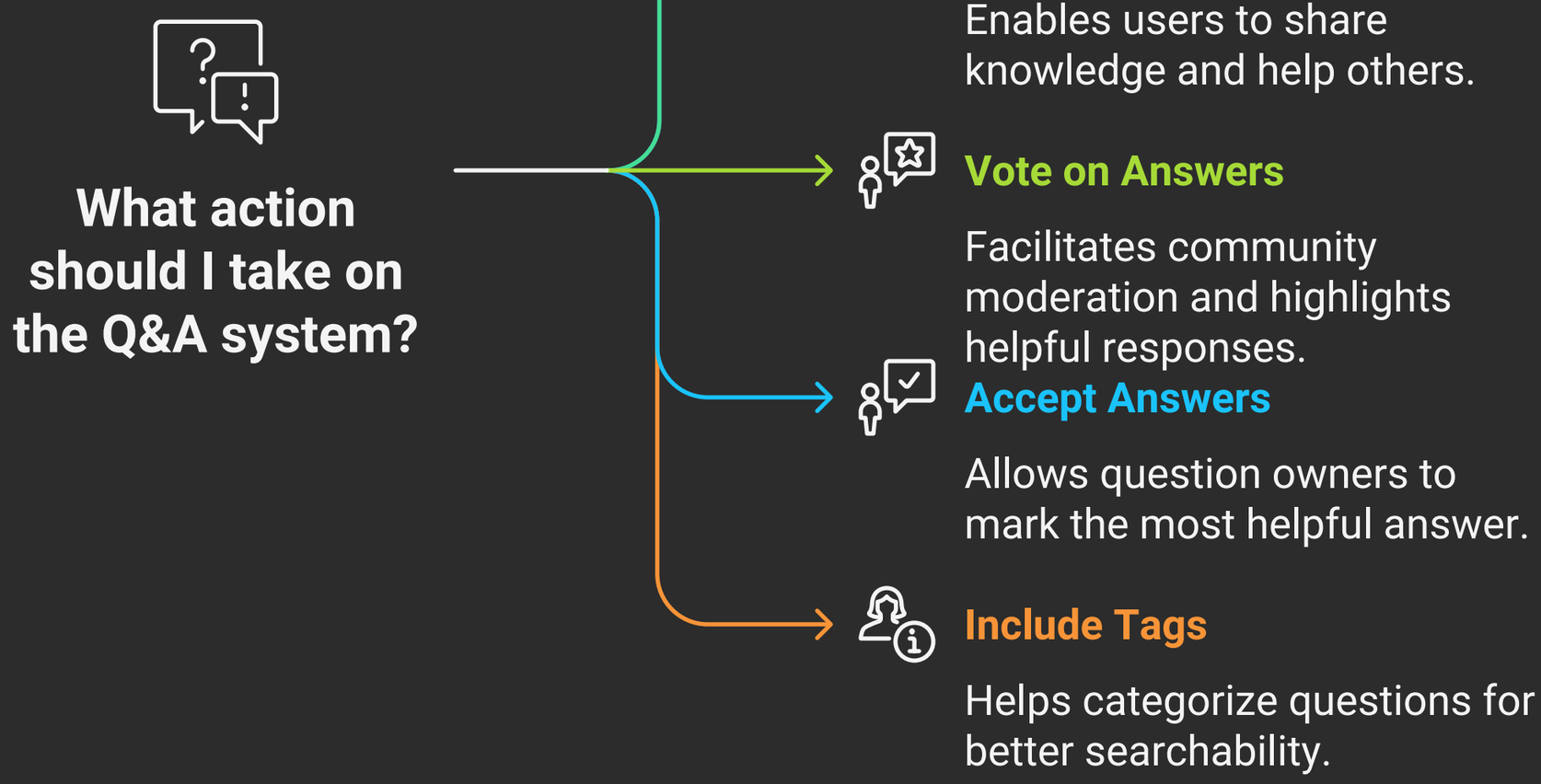
Epic 2: Q&A Core System

StoryTasks

As a user, I can ask questions | - Model: Question(title, desc, tags, owner) - Rich Text Editor (Quill.js) with bold, italic, img, align, etc - POST /questions/ in DRF + React form

As a user, I can answer questions | - Answer model (linked to Question + User) - POST /answers/ endpoint - Answer list on question page

As a user, I can vote on answers | - Vote model (Upvote/Downvote per user/answer) - Button UI with vote count logic
As a question owner, I can accept an answer | - Boolean is_accepted on answer - PATCH logic and styling for selected answer
As a user, I must include tags | - Tag model, multiselect input - M2M link w/ Question



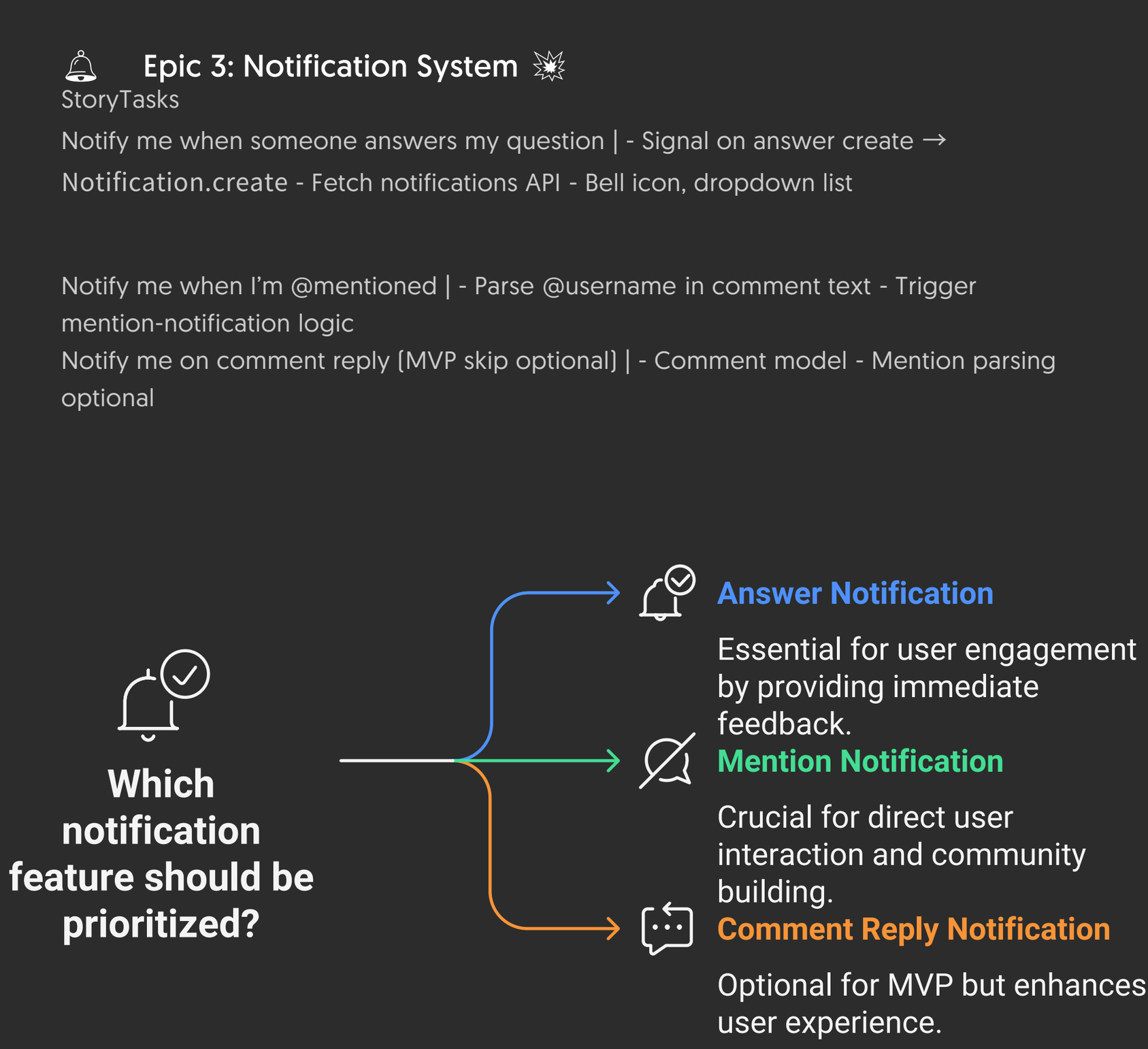
Epic 3: Notification System

StoryTasks

Notify me when someone answers my question | - Signal on answer create → Notification.create - Fetch notifications API - Bell icon, dropdown list

Notify me when I’m @mentioned | - Parse @username in comment text - Trigger mention-notification logic

Notify me on comment reply (MVP skip optional) | - Comment model - Mention parsing optional



Epic 4: ML/DL Integration

StoryTasks

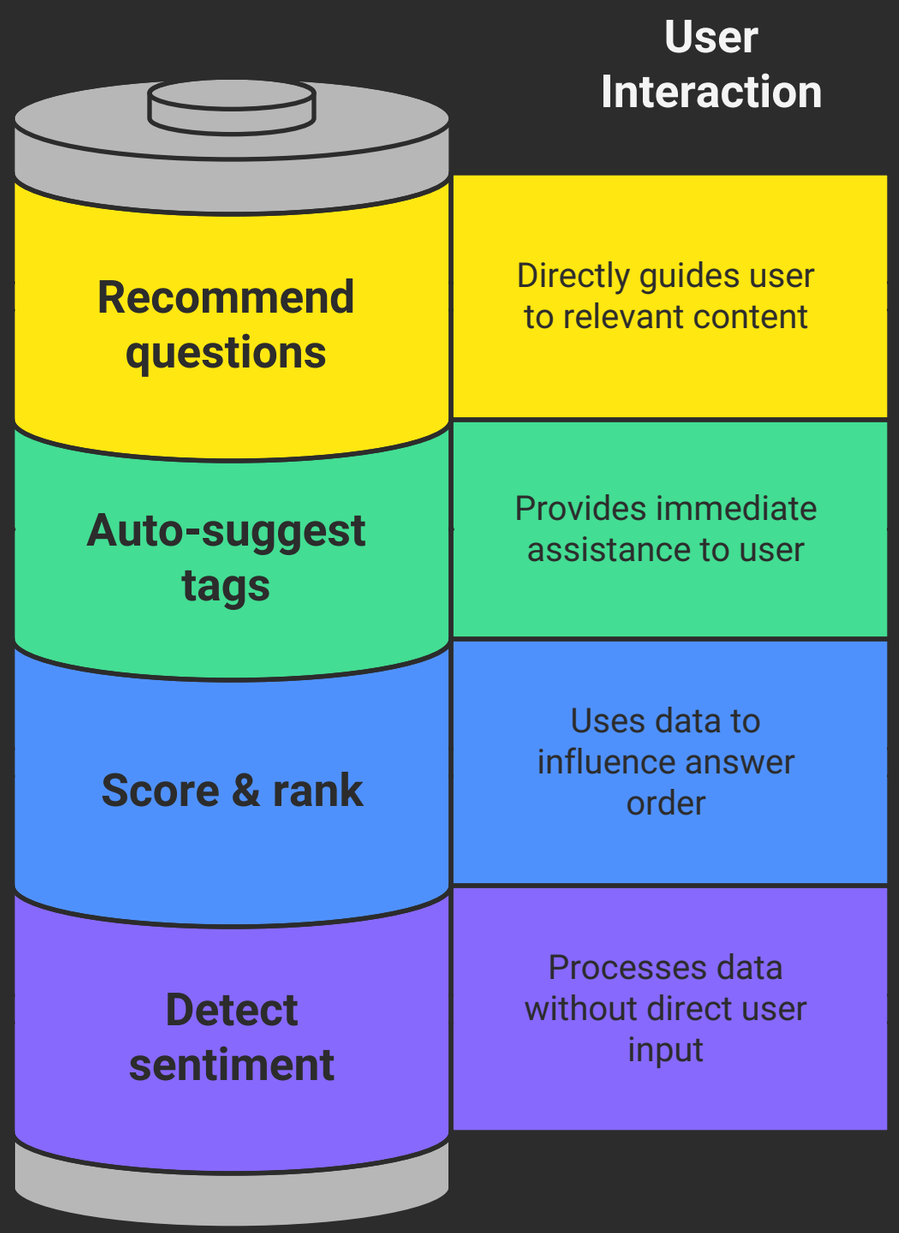
Detect sentiment of answers | - Create Flask/FastAPI: POST /sentiment - Use VADER or Transformers - On answer submit, call API, save sentiment label

Recommend questions to user | - TFIDF recommender based on tags/question text - POST /recommend returns question list - Integrate in homepage

Auto-suggest tags for question | - NER tagging or keyword extraction - /suggest-tags → returns top 5 - Show suggestions in frontend

Score & rank answers | - Heuristic ML model based on upvotes, sentiment, length - Sort answers accordingly

Categorizing tasks by level of user interaction



Epic 5: UI/UX Features

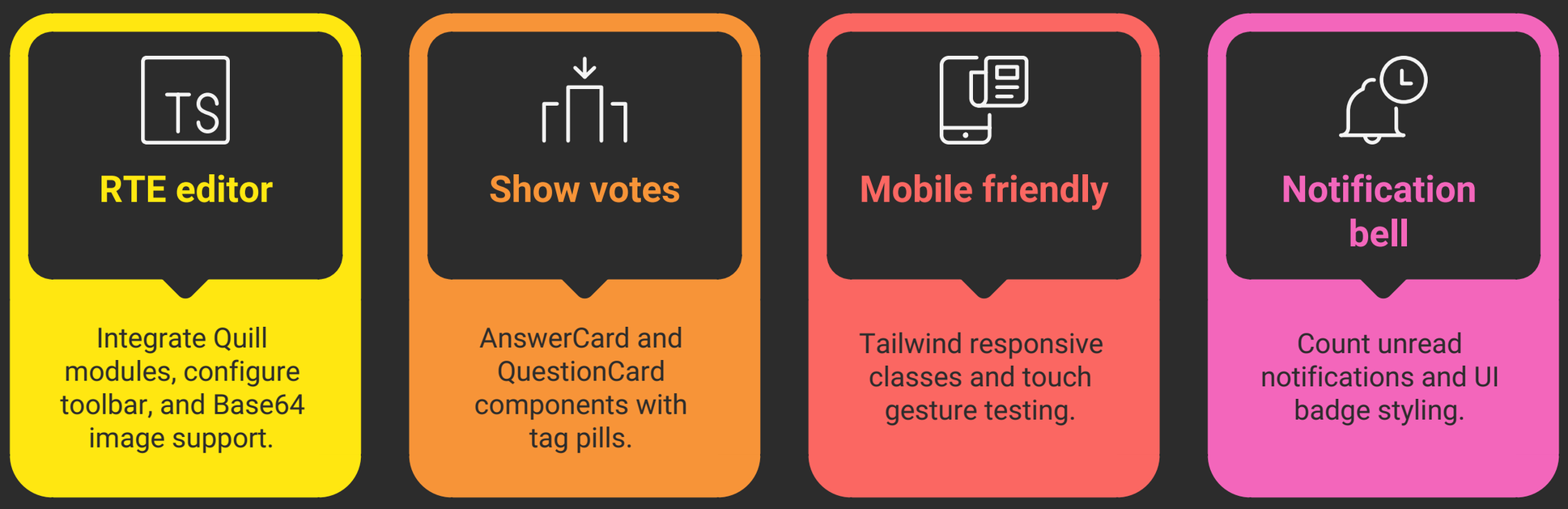
StoryTasks

RTE editor w/ image, links, emoji | - Integrate Quill modules - Configure toolbar - Base64 image support or S3

Show votes, owner, tags per Q/A | - AnswerCard + QuestionCard components - Tag pill components
Mobile friendly layout | - Tailwind responsive classes - Test touch gestures

Notification bell w/ unread count | - Count unread, mark read logic - UI badge styling

StoryTasks Features



Epic 6: DevOps + CI/CD

StoryTasks

Setup Docker for Django + ML | - Dockerfile for Django - Dockerfile + requirements for Flask - docker-compose linking services

Deploy frontend on Netlify | - Auto deploy via GitHub push - Configure env vars

Deploy backend + ML on Render | - Free tier Django - Public ML API on Render

GitHub Actions for tests | - YAML for test/lint run on PRs

Streamlining StackIt's Deployment with Agile Practices

Setup Docker for Django + ML

Dockerfile creation for Django and Flask

Deploy frontend on Netlify

Auto deploy via GitHub push

Deploy backend + ML on Render

Free tier Django and public ML API

GitHub Actions for tests

YAML for test/lint run on PRs