



SoftWare Engineering Project

Humbat Jamalov

Rauf Jabarov

MirSaid Hesenzade

Fakhriyya Huseynova



The Problem: Scheduling Challenges

Scheduling a meeting often turns into a long back-and-forth. It's time-consuming, inefficient, and frustrating for groups. Most teams struggle to quickly find a time that works for everyone.



Solution Overview

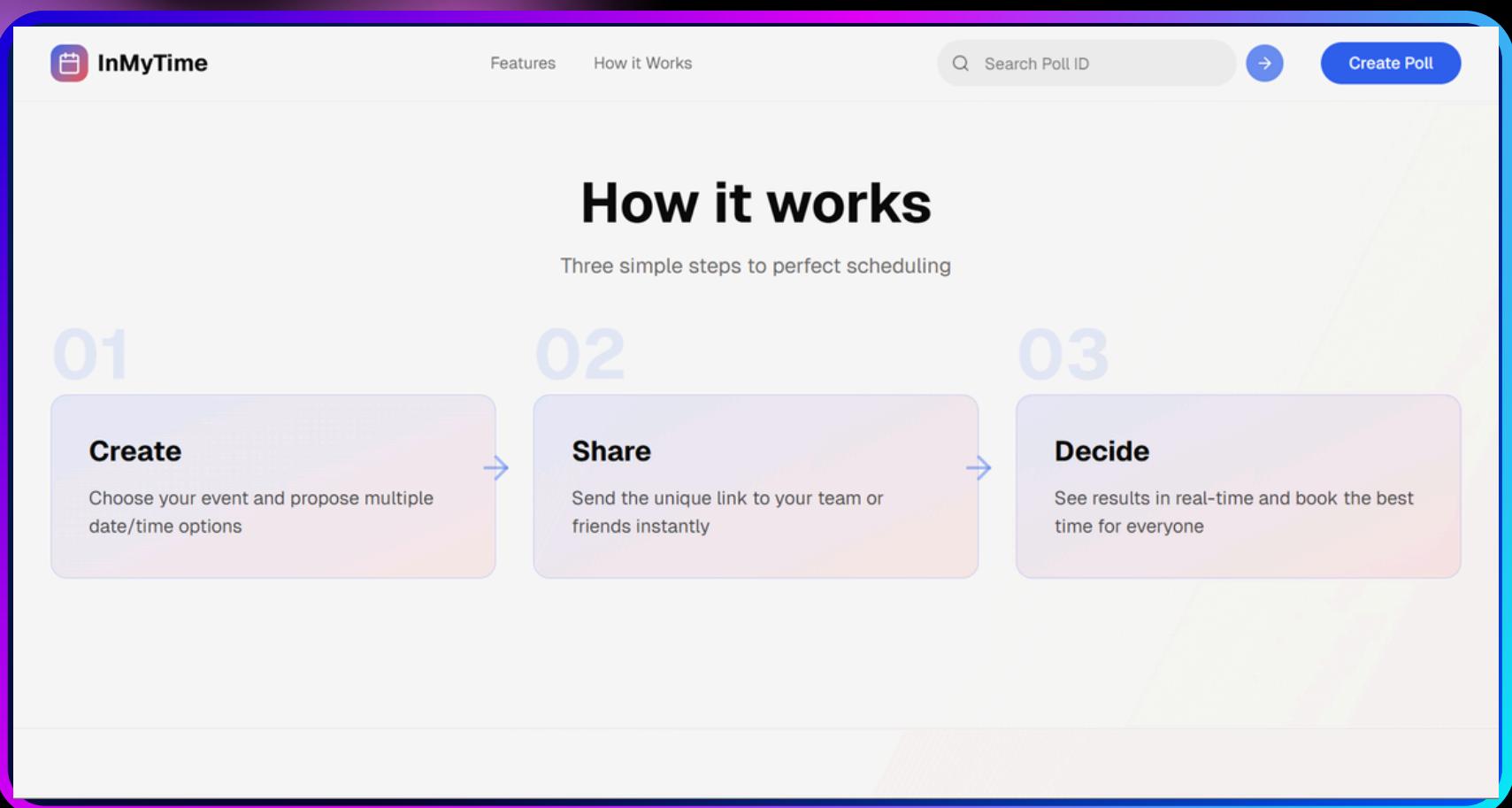
Our solution is a web-based platform that helps groups quickly find a common available time. It collects everyone's availability, visualizes the results, and automatically highlights the best meeting options. This makes scheduling fast, transparent, and effortless.



OUR COMPETITORS

	InMyTime	When2Meet	Doodle
Real-time availability updates			
No Registration Required			
Host-controlled final selection			

Overview of Our Platform

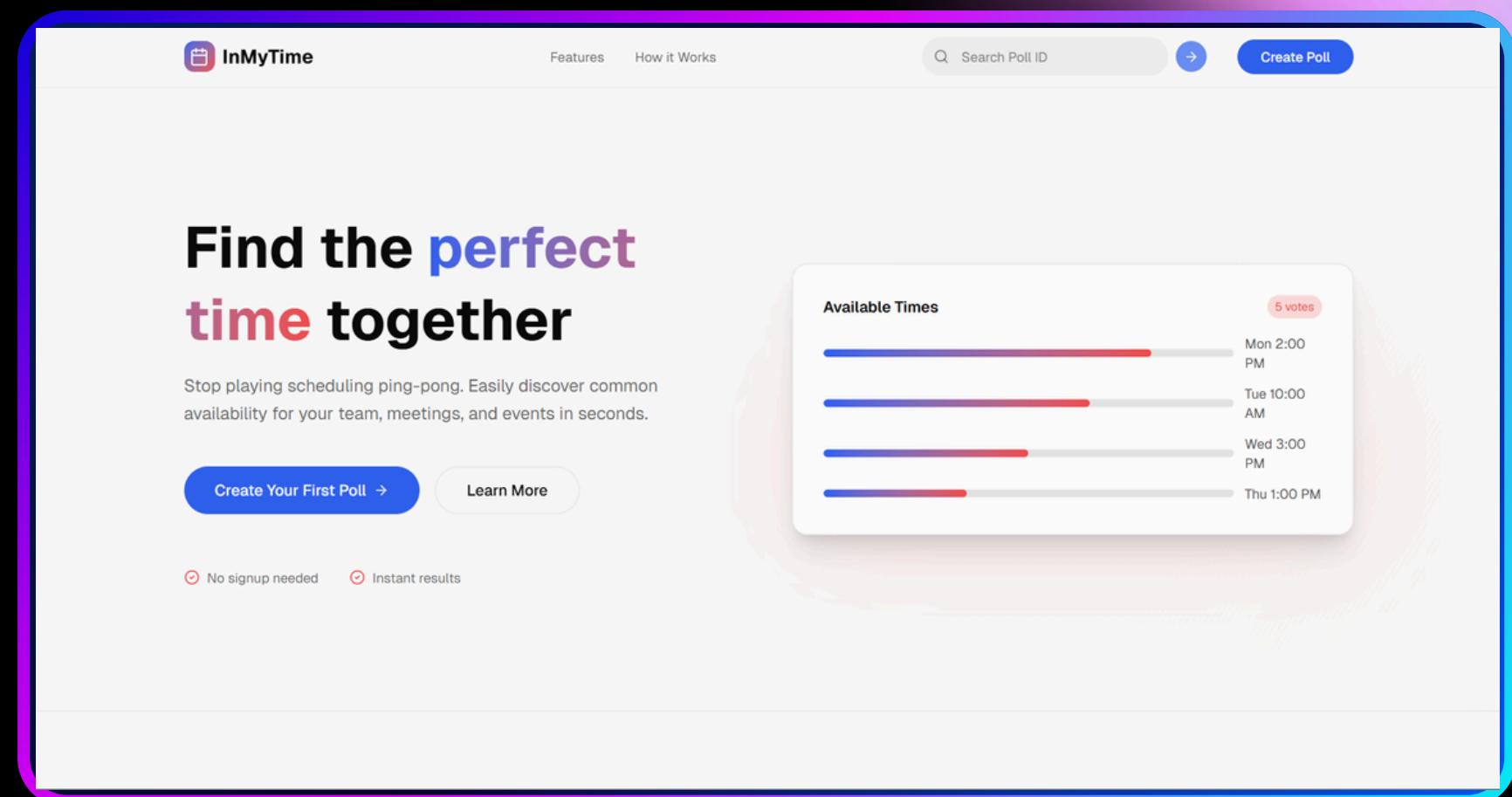


How it works

Three simple steps to perfect scheduling

- 01 Create**
Choose your event and propose multiple date/time options
- 02 Share**
Send the unique link to your team or friends instantly
- 03 Decide**
See results in real-time and book the best time for everyone

Users create a poll, share it with participants, and instantly see the combined availability. The process is simple and intuitive, making scheduling smooth and fast.



Find the perfect time together

Stop playing scheduling ping-pong. Easily discover common availability for your team, meetings, and events in seconds.

[Create Your First Poll →](#) [Learn More](#)

Available Times

Day	Time	Votes
Mon	2:00 PM	5 votes
Tue	10:00 AM	
Wed	3:00 PM	
Thu	1:00 PM	

No signup needed Instant results

The landing page highlights the core value of the platform: finding the perfect meeting time effortlessly. It clearly showcases the main features



Create Poll

The screenshot shows the 'CREATE POLL' section of the InMyTime app. On the left, there's a form with fields for 'Event title' (Product sync, sprint review, study session...), 'Description' (Add an optional note...), and 'Target dates'. Below these are buttons for 'SINGLE DAY' (dd.mm.yyyy) and 'DATE RANGE' (31.12.2025 - 30.01.2026). A '38 selected' badge is visible. To the right is a 'LIVE PREVIEW' titled 'Available Times Overview' showing a list of dates from Mon, Dec 15 to Thu, Jan 8, each with a color-coded bar indicating availability.

Users select dates, set time ranges, and choose slot duration. A live preview helps them understand how the final schedule will look before sharing.

Participants choose their available slots on an interactive grid, with real-time updates, color-coded availability, and clear statistics to guide scheduling decisions.

The screenshot shows the 'VOTING PAGE' for a poll titled 'salam'. It displays a grid of 28 time slots (4 days by 7 hours) with a legend: light green for 'Smal' (1 slot), medium green for 'Popular' (2 slots), and dark green for 'Your Choice' (3 or more slots). The 'POLL DETAILS' sidebar indicates 15m slots, 9 participants, and 26 days. The 'Stats Overview' sidebar shows 79 total votes and Friday, Nov 7 as the best day. The 'Participants' sidebar lists 9 participants.

Voting Page



Finalize Feature

After everyone submits their availability, the host selects the best meeting time and closes the poll. A clear confirmation is shown to all participants, making the final step simple and transparent.

The screenshot shows the 'Group Meet' screen in the InMyTime app. At the top, there's a banner with a trophy icon and the text 'Final Meeting Time Confirmed!' followed by 'This poll is officially closed.' To the right, the time '11:00 AM' and date 'Sunday, November 23' are displayed. Below the banner, it says 'Slot Duration: 30 mins' and 'Total Participants: 13'. A section titled 'Participant Details' shows a list of participants with their initials and names: salam, humbat, Jamal, Sadiq, Matin, Хохан, humba, Rauf, and Smallll. Next to each participant name is a small profile picture and a status indicator showing '6 днк.' (6 days). To the right of the participant list, there are several greyed-out circular icons with numbers indicating available slots: '0 slots', '0 slots', '0 slots', '4 slots', '0 slots', '0 slots', '1 slots', '0 slots', and '0 slots'. The background of the app interface has a pink and purple gradient.



The screenshot shows a task management interface with a sidebar for assignees and a main list divided into three sections: Backlog, In progress, and Done.

Assignees:

- Camalzadeh (8)
- Copilot (2)
- fakhriyyaa (3)
- Rafetikus (5)
- Said2911 (5)

Backlog (3 items):

- Change structure of Base and Home #29
- Change Readme file #39
- Modularize the project structure #49

In progress (2 items):

- Update poll/create screen #17
- Add presentation of our project #40

Done (16 items):

- Create new database architecture #8
- Create well designed poll/create screen #6
- Create model integration tests #12
- Create unit tests #11
- Update design of poll/[id] screen #9
- Update Readme file #18
- Update Project Version #48
- Search by Id at home screen #28
- Fix current app issues #33
- Add Unit tests #37
- Add Websocket for realtime communication #16
- Create API integration tests #14
- Set limits at the backend part also #27
- Update home screen #24
- Add Integration Test #38
- Fix whole project #41

DEVELOPMENT WORKFLOW

✓ Task Management

- GitHub Issues used to track features and bugs

🔧 Branching Strategy

- Feature branches kept main branch stable

📝 Pull Requests

- Code reviewed before merging

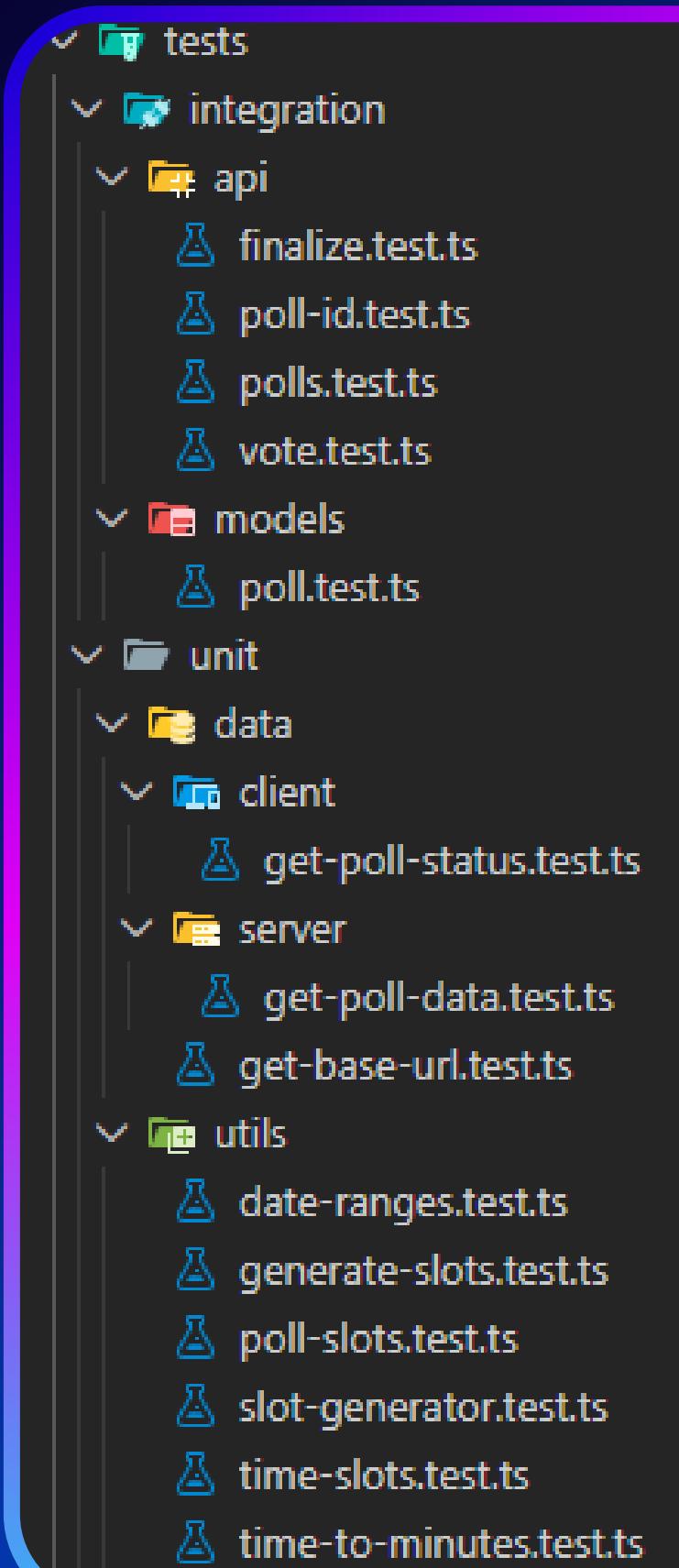
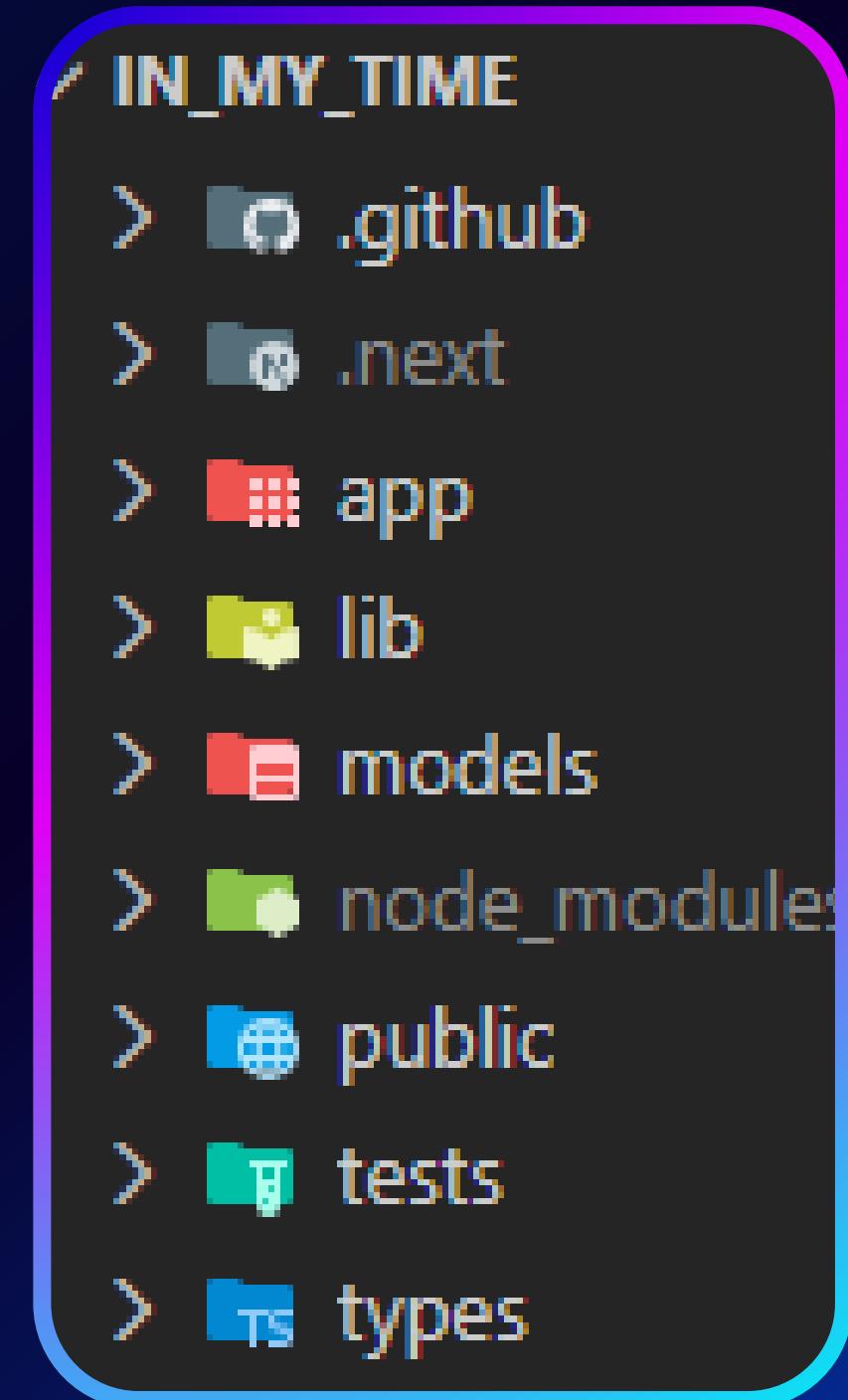
🧪 Testing

- Integration tests run before each merge



Project Structure

Our project is organized into clear modules: **app** for pages and API routes, **components** for reusable UI blocks, **lib** and **utils** for shared logic, **models** for data schemas, **types** for TypeScript definitions, and **tests** for automated testing.





Application Architecture

The project uses TypeScript across both frontend and backend. The frontend is built with Next.js and React, while the backend uses Next.js API routes with MongoDB and Mongoose. Jest is used for all testings.

Technology Stack & Tools

(What Powers InMyTime)





Security Update: React RSC Vulnerability Fix

Why we updated

- A critical vulnerability was discovered in React Server Components (CVE-2025-55182)
- Affected version: 19.2.0
- The issue allowed unauthenticated remote code execution

Link for more information

- <https://react.dev/blog/2025/12/03/critical-security-vulnerability-in-react-server-components>
- <https://nextjs.org/blog/CVE-2025-66478>

What we changed

- Updated React RSC package from 19.2.0 → 19.2.1

Impact on our project

- Our app is now protected against the vulnerability
- No changes to UX/functionality
- Update required only back-end build dependencies



InMyTime

Page 12

Thank you for your attention.

With InMyTime, scheduling becomes faster, clearer, and more collaborative – helping teams stay organized and aligned.

We look forward to improving and expanding the platform together.

When you face challenges with time, the answer is in my time.

InMyTime

-

Our GitHub